

**TIMING OF THUNDERSTORM  
OCCURRENCE FOR CAPE CANAVERAL,  
FLORIDA**

**THESIS**

**Thomas G. Renwick, Captain, USAF**

**AFIT/GM/ENP/00M-10**

**DEPARTMENT OF THE AIR FORCE  
AIR UNIVERSITY**

***AIR FORCE INSTITUTE OF TECHNOLOGY***

---

**Wright-Patterson Air Force Base, Ohio**

**APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.**

**DTIC QUALITY INSPECTED 4**

**20001113 023**

The views expressed in this thesis are those of the author, and do not reflect the official policy or position of the Department of Defense, or the U.S. Government.

AFIT/GM/ENP/00M-10

TIMING OF THUNDERSTORM OCCURRENCE FOR CAPE CANAVERAL,  
FLORIDA  
THESIS

Presented to the Faculty

Department of Engineering Physics

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Masters of Science in Meteorology

Thomas G. Renwick, B.S.

Captain, USAF

March 2000


APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

TIMING OF THUNDERSTORM OCCURRENCE FOR CAPE CANAVERAL,  
FLORIDA

Thomas G. Renwick, B.S.

Captain, USAF

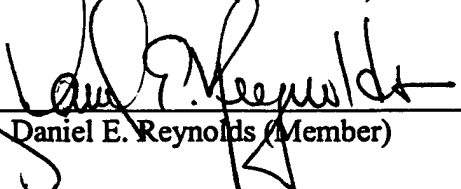
Approved:



Cecilia A. Miner (Chairman)

29 Feb 2000

Date



Daniel E. Reynolds (Member)

29 Feb 2000

Date



Michael K. Walters (Member)

29 Feb 2000

Date

## ACKNOWLEDGEMENTS

First I would like to thank my committee members, Lieutenant Colonel Cecilia Miner, Lieutenant Colonel Michael Walters, and Professor Daniel Reynolds whose expertise in meteorology and statistics guided me throughout my entire research. Without their help, this thesis would not be as complete or as scientifically sound.

I would also like to thank my friends and fellow classmates for their unending patience and support. They were always willing to listen to my ideas and suggest ways of solving them. The advice of Captains Jim Trigg, Michael Calidonna, Steven Dickerson and First Lieutenant Jon Saul helped me from wasting an incalculable amount of work time. I'd also like to thank Mrs. Vlasta L. Renwick for her expert editorial advice. I especially want to thank my brother, Mr. Ian J. Renwick, for his computer programming and spreadsheet skills. His intimate knowledge of data management and computer programming helped in taking an enormous amount of data and being able to put it in a more useable format.

Finally, I would like to thank all my instructors who taught me throughout the year. Without that knowledge, this thesis would have suffered greatly.

Thomas G. Renwick

## TABLE OF CONTENTS

<b>ACKNOWLEDGEMENTS.....</b>	<b>ii</b>
<b>LIST OF FIGURES.....</b>	<b>vii</b>
<b>LIST OF TABLES.....</b>	<b>x</b>
<b>LIST OF ACRONYMS.....</b>	<b>ix</b>
<b>ABSTRACT.....</b>	<b>xi</b>
<b>1. Introduction.....</b>	<b>1</b>
1.1 OVERVIEW .....	1
1.2 BACKGROUND .....	3
<i>1.2.1 Thunderstorms.....</i>	<i>3</i>
<i>1.2.2 Neumann-Pfeffer Thunderstorm Index (NPTI) and Timing Scheme.....</i>	<i>5</i>
1.3 STATEMENT OF PROBLEM.....	6
<i>1.3.1 Objectives .....</i>	<i>7</i>
<i>1.3.2 Scope .....</i>	<i>7</i>
<i>1.3.3 Benefit of Solving the Problem.....</i>	<i>8</i>
1.4 PROCEDURE .....	9
1.5 SUMMARY OF RESULTS .....	10
1.6 OUTLINE OF THESIS .....	10
<b>2. Literature Review .....</b>	<b>11</b>
2.1 SEA BREEZE FORMATION .....	11

2.2 SYNOPTIC WIND FLOW .....	12
2.2.1 Type I days .....	12
2.2.2 Type II days .....	13
2.2.3 Type III days.....	14
2.3 SEA BREEZE CIRCULATION MODELS .....	16
2.3.1 Model Run for Type I Days.....	16
2.3.2 Model Run for Type II days.....	17
2.3.3 Model Run for Type III days .....	17
2.4 BACKGROUND WORK .....	18
2.4.1 Frequency and Duration of Thunderstorms at Cape Kennedy Part I.....	19
2.4.2 Frequency and Duration of Thunderstorms at Cape Kennedy, Part II .....	21
2.4.3 Thunderstorm Forecasting at Cape Kennedy, Florida, Utilizing Multiple Regression Techniques .....	21
2.4.4 Thunderstorm Starting Time .....	24
<b>3. Methodology .....</b>	<b>26</b>
3.1 DATA USED .....	26
3.2 INTERPOLATION.....	28
3.3 COMPUTATIONS.....	29
3.3.1 Mean Relative Humidity.....	29
3.3.2 Showalter Stability Index .....	30

3.4 TYPES OF REGRESSION .....	30
3.4.1 REEP.....	31
3.4.2 Logistic Regression .....	32
3.5 RESEARCH APPROACH.....	33
3.5.1 Neumann Probability (NP) and Neumann Timing (NT) .....	33
3.5.2 Logistic Regression Probability (LRP) and Neumann Timing (NT).....	34
3.5.3 NP and Linearly Regressed Timing Coefficients (LRTC).....	35
3.5.4 Logistic Regression Probability (LRP) and Linearly Regressed Timing Coefficients (LRTC) .....	36
3.5.5 Average Thunderstorm Starting Time.....	36
3.6 VERIFICATION DATA SET .....	36
<b>4. Statistical Analysis and Results .....</b>	<b>38</b>
4.1 JUNE TIMING SCHEMES.....	38
4.2 JULY TIMING SCHEMES.....	44
4.3 AUGUST TIMING SCHEMES .....	47
4.4 SEPTEMBER TIMING SCHEMES .....	50
<b>5. Conclusions and Recommendations.....</b>	<b>54</b>
5.1 CONCLUSIONS.....	54
5.2 RECOMMENDATIONS .....	55
5.3 SUGGESTIONS FOR FURTHER RESEARCH.....	55

<b>Appendix A. Constants for NPTI.....</b>	<b>57</b>
<b>Appendix B. Neumann Time Coefficients.....</b>	<b>58</b>
<b>Appendix C. Interpolation Program using IDL®.....</b>	<b>59</b>
<b>Appendix D. Qbasic® Program Finds Nearest 50 mb Increment.....</b>	<b>62</b>
<b>Appendix E. IDL® Program Used to Find Mean RH.....</b>	<b>66</b>
<b>Appendix F. IDL® Program Used to Find SSI.....</b>	<b>70</b>
<b>Appendix G. Mathcad® Template to Find NPTI.....</b>	<b>75</b>
<b>Appendix H. Logistical Probability Coefficients.....</b>	<b>81</b>
<b>Appendix I. Mathcad® Template to Find LRP with NT.....</b>	<b>82</b>
<b>Appendix J. Linearly Regressed Time Coefficients.....</b>	<b>88</b>
<b>Appendix K. Mathcad® Template to Find NP with LRTC.....</b>	<b>89</b>
<b>Appendix L. Linearly Regressed Timing Coefficients using Logistically Regressed Probabilities.....</b>	<b>92</b>
<b>Appendix M. Mathcad® Template to Find LRP with LRTC.....</b>	<b>93</b>
<b>Appendix N. Mathcad® Template for Random Verification Set.....</b>	<b>96</b>
<b>BIBLIOGRAPHY.....</b>	<b>97</b>
<b>VITA.....</b>	<b>99</b>

## LIST OF FIGURES

Figure	Page
1. Map of Florida.....	2
2. Map of Cape Canaveral.....	4
3. Mean Synoptic Wind Field for Type I Days.....	13
4. Mean Synoptic Wind Field for Type II Days.....	14
5. Mean Synoptic Wind Field for Type III Days.....	15
6. Neumann 15-Day Moving Average Using 15 Year Data Set.....	20
7. Graph of Mean and Standard Deviation from Actual Starting Times for June.....	41
8. Box and Whiskers of Error Data for June.....	43

9. Graph of Mean and Standard Deviation from Actual Starting Times for July.....	45
10. Box and Whiskers of Error Data for July.....	46
11. Graph of Mean and Standard Deviation from Actual Starting Times for August.....	48
12. Box and Whiskers of Error Data for August.....	49
13. Graph of Mean and Standard Deviation from Actual Starting Times for September.....	51
14. Box and Whiskers of Error Data for September.....	52

## LIST OF TABLES

Table	Page
1. Available Surface Observations.....	26
2. Available Upper Air Observations.....	26
3. Years with Available Surface Observations and Upper Air Observations.....	27
4. Values of Showalter Stability Index and Descriptions.....	30
5. Mean and Standard Deviations of Error Between Actual Starting Times and Forecast Starting Times of June.....	38
6. Mean and Standard Deviations of Error Between Actual Starting Times and Forecast Starting Times of July.....	44
7. Mean and Standard Deviations of Error Between Actual Starting Times and Forecast Starting Times of August.....	47
8. Mean and Standard Deviations of Error Between Actual Starting Times and Forecast Starting Times of September.....	50

## **LIST OF ACRONYMS**

**AFB** – Air Force Base

**AFCCC** – Air Force Combat Climatology Center

**AWS** – Air Weather Service

**ECSB** – East Coast Sea Breeze

**EST** – Eastern Standard Time

**IDL** – Interactive Data Language

**LRP with LRTC** – Logistically Regressed Probability with Linearly Regressed Timing  
Coefficients

**LRP with NT** – Logistically Regressed Probability with Neumann Timing

**NP with LRTC** – Neumann Probability with Linearly Regressed Timing Coefficients

**NP with NT** – Neumann Probability with Neumann Timing

**NPTI** – Neumann-Pfeffer Thunderstorm Index (equivalent to **NP with NT**)

**PDF** – Probability Density Function

**REEP** – Regression Estimation of Event Probabilities

**SSI** – Showalter Stability Index

**UTC** – Universal Time Coordinated

**WCSB** – West Coast Sea Breeze

**WS** – Weather Squadron

## **ABSTRACT**

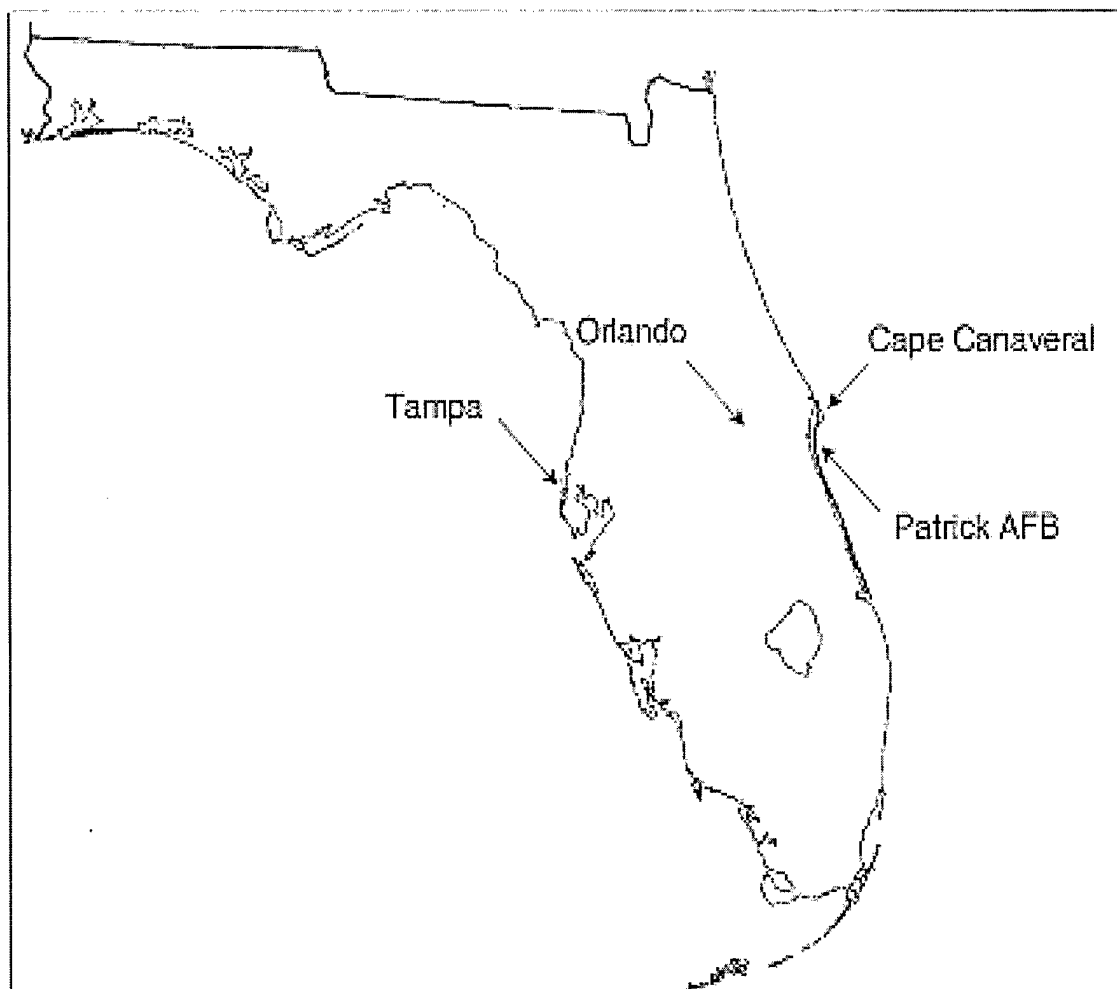
This research is concerned with improving an existing algorithm to accurately forecast thunderstorm starting times for Cape Canaveral, Florida. This was accomplished by investigating different linear regression techniques than those used in the existing algorithm. The result is three new thunderstorm start time algorithms. The forecast start times of these new algorithms were then compared to actual thunderstorm start times to determine which method produced the most accurate results. The average thunderstorm starting time was also calculated from the data. This time was also compared to actual thunderstorm starting time. Upon examination of the various start times produced, it was found that all algorithms, including the original algorithm, performed worse than using the average thunderstorm start time.

## **1. Introduction**

Many thunderstorm algorithms have been created that give an overall probability of a thunderstorm occurring on a given day. This is useful information for both meteorologists and the average person. One aspect that always tends to be overlooked, however, is the starting time of the thunderstorm occurrence. That is, given that a thunderstorm is expected to occur on a given day, what time will it occur? Generally, a vague notion of early morning or late afternoon is given but this is hardly scientific and gives the impression that the meteorologist is taking a "best-guess." Not only will knowledge of the timing of the thunderstorm maximize safety for individuals working outside, it will also reduce costs to weather sensitive operations. Therefore, a review of the current techniques to find thunderstorm timing with the intent of improving accuracy can have an important effect on operations, especially if an even more accurate timing scheme can be applied.

### **1.1 Overview**

The 45<sup>th</sup> Weather Squadron (WS), located at Patrick Air Force Base (AFB) in Florida is responsible for forecasting all weather phenomena at Patrick and for supporting operations at Cape Canaveral, Florida. Not only is the squadron responsible for flight weather briefings for pilots, but they also produce launch weather forecasts for the launch weather officers located on the Cape who deal with satellite and shuttle launches. Obviously, weather plays an important role in many operations underway at Patrick AFB and the Cape. See Figure 1 for the geographic location of Cape Canaveral (Kennedy Space Center) in Florida.



**Figure 1 Map of Florida**

Of particular interest to the 45<sup>th</sup> WS is if and when a thunderstorm will occur.

Currently, the 45<sup>th</sup> WS has one method to estimate the start time of a thunderstorm occurring on station. This method is the Neumann-Pfeffer Thunderstorm Index (NPTI) which was created in 1971 and has recently been shown to have some inaccuracies (Howell 1998). More recently, this index has been improved by examining different constants and regression techniques (Everitt 1999). As of this writing, Everitt's new index is not operational because all programming was performed in Mathcad<sup>®</sup>. The

forecasters at Patrick AFB do not have access to Mathcad<sup>®</sup> nor do they know how to use it. The NPTI calculates the probability of a thunderstorm occurring on a given day and also claims a starting time with an error factor of  $\pm 1 \frac{1}{2}$  hours when given 5 inputted values (Neumann 1971). This starting time is reported at 1100UTC and is valid that same day. This thesis will focus on finding a better algorithm for thunderstorm starting time with better error factors.

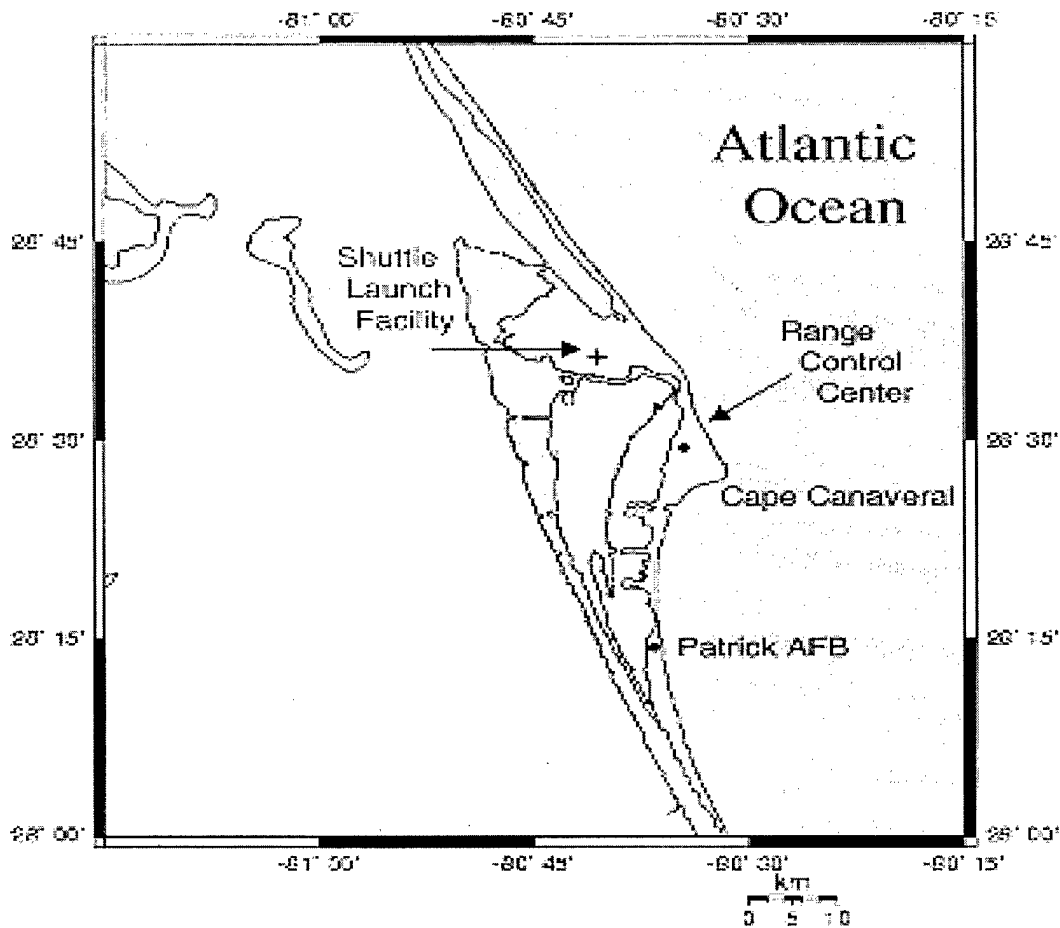
## **1.2 Background**

Thunderstorms play an important role in the day-to-day operations at the 45<sup>th</sup> WS. Thunderstorms restrict all flight operations, maintenance personnel and ground operations at Patrick AFB and Cape Canaveral. Unfortunately the weather squadron is located in the state that has the highest concentration of thunderstorms in the nation. Falls *et al.* (1971), Byers and Rodebush (1948), and many others have also commented that the area where Patrick AFB is located is subject to one of the highest frequencies of thunderstorms in the world. Obviously, the incidence and timing of thunderstorms is of utmost importance to all individuals concerned with operations that are weather sensitive at Patrick AFB and the Cape.

### **1.2.1 Thunderstorms**

In order for a thunderstorm to occur, three ingredients are necessary: moisture, instability, and lift. Florida has all of these ingredients in abundance. First, Florida is a peninsula and as such is nearly completely surrounded by water and has ample moisture.

To make matters worse, Patrick AFB is surrounded by rivers which increase the amount of moisture in the area. Figure 2 shows the geographic location of Patrick AFB and Cape



**Figure 2 Map of Cape Canaveral**

Canaveral and available moisture for thunderstorm formation. Secondly, Florida is located far south enough that the instability caused by the warm summer temperatures is further enhanced by the subtropics. Therefore, the instability over this region is highly conducive for thunderstorm formation. Finally, lift is also abundant in Florida. Synoptic

lifting along with a meso-scale trigger has long been known to cause thunderstorms in the mid-latitudes. Byers and Rodebush (1948) determined that these synoptic features generally do not reach far enough south in Florida during summer to cause this lift. However, it has been determined that the necessary lift can be supplied by the sea-breeze (Byers and Rodebush 1948; Gentry and Moore 1954; Frank *et al.* 1967; Pielke and Cotton 1977; Burpee and Lahiff 1984; Blanchard and Lopez 1985 and many others). This interaction of the sea breeze and the local environment becomes the largest predictor of lift and also the timing of thunderstorm occurrence (Byers and Rodebush 1948).

### **1.2.2 Neumann-Pfeffer Thunderstorm Index (NPTI) and Timing Scheme**

The NPTI is the current tool used by the 45<sup>th</sup> WS to forecast thunderstorm probability and starting time. It was created in 1971 by Charles Neumann, and it calculates the probability of a thunderstorm occurring at Cape Canaveral and also reports a time that the thunderstorm can be expected on station. In order to find the thunderstorm starting time, the probability of a thunderstorm occurring must first be calculated. Neumann's probability index uses the day number, the 850 mb and 500 mb orthogonal wind components, the 800 mb to 600 mb mean relative humidity, and the Showalter Stability Index (SSI) to come up with the probability of a thunderstorm occurring. The above variables are individually regressed linearly by month against the dependent variable, daily thunderstorm occurrence. In essence, the first regression produces a probability of thunderstorm occurrence when only considering one variable. This probability of thunderstorm occurrence is then placed into another equation and again linearly regressed

against thunderstorm occurrence. The final result is a probability of thunderstorm occurrence with all five parameters being considered. The data set used in the regressions encompasses the years from 1950 – 1951 and 1957 – 1969. Once the thunderstorm probability has been calculated it is used to find the starting time. For Neumann's data set, he calculated the average starting time of thunderstorms over the Cape and after assuming normality, calculated the standard deviation. He found that if the orthogonal wind components at 850 mb, forecast probability of thunderstorm occurrence and day number were considered, then the standard deviation for thunderstorm starting time could be predicted and a starting time could be deduced.

### **1.3 Statement of Problem**

How can thunderstorm timing accuracy be increased for Cape Canaveral, Florida? The first way to increase timing accuracy is to increase the accuracy of the thunderstorm probability. Once an increase in accuracy for thunderstorm probability has been attained, the new probability may be useful in producing better times.

This thesis will report on four different methods of improving this timing. First, a logistic regression to find the probability of thunderstorm occurrence is examined instead of using a linear regression (Everitt 1999). The new probability of thunderstorm occurrence is then applied using Neumann's timing scheme to see if it produces more accurate results. The second technique requires using Neumann's original method to determine the probability of thunderstorm occurrence and performing a new linear regression for the start times. Next, logistic regression to find the probability of thunderstorm occurrence is used in conjunction with the new linear regressions to

produce start times. Finally, the average thunderstorm start time is calculated and the difference in time between the average and actual thunderstorm start times is calculated.

### **1.3.1 Objectives**

The purpose of this thesis is to develop a method to improve the timing forecast for thunderstorms at Patrick AFB and Cape Canaveral. The goal is to create an algorithm with improved timing accuracy over that which is being used currently. Because Everitt (1999) found that logistic regression increases the hit rate of thunderstorm occurrence by 17%, it appears that using this probability technique in Neumann's timing scheme could increase the accuracy. This increased accuracy would lead to better forecasts, and all parties stationed at Patrick AFB and Cape Canaveral concerned with weather effects would benefit from this knowledge.

### **1.3.2 Scope**

This thesis will be limited to the study of summer thunderstorm timing at Cape Canaveral, Florida. Thunderstorm timing, from 1950 to 1998, recorded by the official observations is used as the dependent variable in several statistical analyses. This research will only examine thunderstorms that occurred in the convectively active season which is defined here as the months from May to September. This period also corresponds to the period used both in Howell's and Everitt's studies (Howell 1998; Everitt 1999).

One important aspect of this study is that the 45<sup>th</sup> WS forecast is issued at 1100UTC so that any upper air data examined after this time is irrelevant. Thus, only upper air data

before 1100UTC which occurs on a thunderstorm day is examined. The upper air data is used to produce probabilities of thunderstorm occurrence and new time coefficients.

Eighty percent of the data is used to produce the probabilities and new linear regressions, while the remaining twenty percent is withheld and used for verification purposes. The verification data consists of randomly selected days extracted from each summer month. The forecast is valid from 0700L – 2400L.

### **1.3.3 Benefit of Solving the Problem**

During every summer season at Patrick AFB, thunderstorms play an important role in most operations. All flights and launches must be cancelled and personnel must leave the flight line and launch sites when thunderstorms are in the area. This causes both delays and costly man-hours lost. An increased accuracy in forecasting the timing of a thunderstorm on station can reduce the number of delays and cancellations. For example, a flight that is expected to occur in the afternoon when a thunderstorm is expected on station can possibly be moved to earlier in the day or to another day without any of the aforementioned people being affected. Patrick AFB is also responsible for launch forecasts for Cape Canaveral. Once again, thunderstorms will delay any and all launches. If, however, the timing of a thunderstorm is known beforehand, then the individuals responsible for launching these vehicles can possibly change the take-off time to one when no thunderstorm is forecast. Roeder (Personal Communication, 1998) has estimated that it costs \$1 million just to de-fuel then prepare the space shuttle again after a thunderstorm is forecast. Obviously, an improved timing scheme will give the

operational team a better chance to tailor all flights and launches to avoid expected thunderstorms.

#### **1.4 Procedure**

This thesis involved three main tasks: data collection and manipulation, regression, and verification. The first task, data collection and manipulation, was the most time consuming and also the most important. Upon receiving the data from the Air Force Climatology Center (AFCCC), the surface observations were matched up with the corresponding upper air data for the same day and month using Microsoft® Access. This upper air data was then examined for all days with data that corresponded to 1100UTC or earlier. If the upper air data had values only after 1100UTC then they were removed from the data set, as was the surface observation. The upper air data then had to be examined to determine if all reported values were present. Unfortunately, the upper air data had missing values which were needed to calculate some of the inputted parameters. Therefore, these values were interpolated. Finally, the interpolated upper air data was used to calculate the values that were needed in the NPTI and timing scheme.

Modifying the type of regression was the second task. Neumann used linear regressions to determine the probability of thunderstorm occurrence and also to find the timing coefficients in his timing scheme. It has been found that using logistic regression to find the probability of thunderstorm occurrence produces better results (Everitt 1999). Logistic regression was used applying the same predictor variables as Neumann, and then this new probability result was used in Neumann's timing scheme. Furthermore, Neumann's timing scheme used linearly regressed variables that he deemed necessary for

accurate timing. A new linear regression was performed with a larger data set to produce new timing coefficients. Finally, the logistic probability regression was used in conjunction with the newly regressed time coefficients to come up with a new thunderstorm starting time. Another method to examine thunderstorm start times is to compare the average thunderstorm start time to actual thunderstorm start times.

The last task, verification, determined which method should be used operationally. By comparing the means and standard deviations from all five methods to the actual starting time of a thunderstorm, it was seen that the average thunderstorm outperformed the new algorithms as well as the NPTI.

### **1.5 Summary of Results**

The NPTI, three forecast start time algorithms and the average thunderstorm start time were used to forecast thunderstorm time on station. These results were then compared to the actual thunderstorm start time. A mean and standard deviation of timing error was produced. Two hundred sixty eight random, independent events were used as the verification set. All algorithms performed worse than using the average thunderstorm starting time.

### **1.6 Outline of Thesis**

A review of relevant literature on this subject can be found in chapter 2. Chapter 3 expands discussion of the data and analysis techniques, followed by a complete discussion of methodology in chapter 4. Chapter 5 presents results, conclusions, and suggestions for future research.

## **2. Literature Review**

Many experiments have been performed over the Florida peninsula to determine what factors contribute to thunderstorm development. One important factor that has been found to cause them is the formation of the sea breeze. The sea breeze can give an insight as to when a thunderstorm can be expected to occur because these two phenomena are related.

### **2.1 Sea Breeze Formation**

The sea breeze is a well-known meteorological phenomenon. The sea breeze forms between landmasses and water and is caused by diurnal solar heating and radiation of the land. As the solar radiation strikes both land and water, they heat up. However, since the thermal capacity of water is much greater than that of land, the land will heat up faster. As the ground warms up, the air will rise and be replaced by cooler air from the adjacent water. This rising air will then move back over the water and sink. This transverse circulation will cause an on-shore sea breeze during the day and an offshore land breeze at night (Reed 1979).

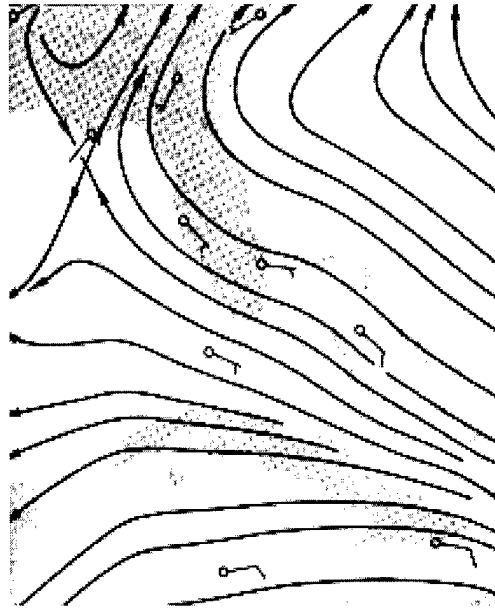
This vertically rising air, if strong enough, can cause thunderstorms. Therefore, the formation of sea breezes, resulting convergence, and the accompanying vertical motions play an important role in thunderstorm development when close to a large water source. The Florida peninsula is an excellent example to study since it is surrounded by the Atlantic Ocean and the Gulf of Mexico.

## **2.2 Synoptic Wind Flow**

Blanchard and Lopez (1985) postulated that several factors are responsible for the daily fluctuations in sea breeze circulations. However, the consensus of opinion is that the most important factor is the synoptic wind flow (Estoque 1962; Nicholls *et al.* 1991; and others). Blanchard and Lopez (1985) further hypothesized that different synoptic regimes can cause discrete temporal and spatial patterns of convection. Therefore, the sea breeze timing, intensity, motion, and accompanying convection should be dependent on the synoptic scale wind flow. In order to see if this was true, Blanchard and Lopez (1985) examined a large rainfall data set. This data set was then compiled to produce a composite rainfall data set. Upon examination of this composite rainfall data, it was apparent that three distinct types of days of rainfall were prevalent over Florida during summer months. For clarity, they are called Type I, Type II, and Type III days.

### **2.2.1 Type I days**

Type I days occur when Florida is under the influence of the Atlantic high. The accompanying synoptic wind flow is from the southeast and is usually weak (Figure 3). As the East Coast Sea Breeze (ECSB) sets up, convection forms along this boundary in the early afternoon, and the general wind flow moves the sea breeze inland. The West Coast Sea Breeze (WCSB) also sets up but moves inland more slowly since the opposing synoptic wind flow partially cancels out the effect of the on-shore flow. The ECSB moves further inland while the WCSB slowly moves inland from the other direction and they finally merge over the western-central interior of the peninsula. Since the merging of



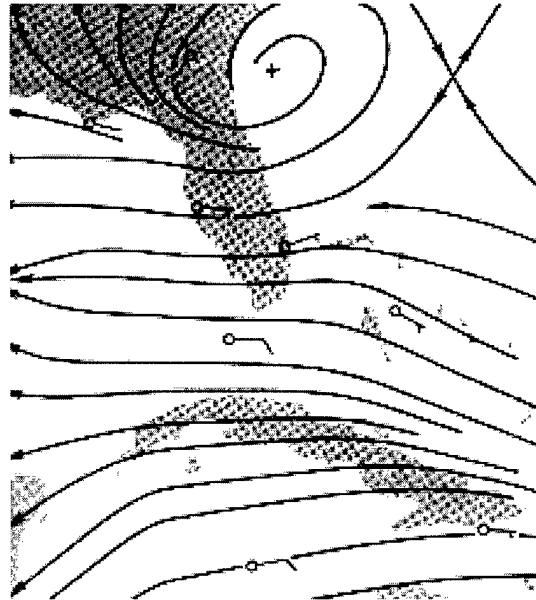
**Figure 3 Mean Synoptic Wind Field for Type I days (Blanchard and Lopez 1985)**

both sea breezes enhances vertical motion, strong convection takes place in this region and thunderstorms develop (Blanchard and Lopez 1985).

### **2.2.2 Type II days**

Type II days are caused by a continental high which is generally located over the southeastern United States and causes easterly synoptic winds over Florida (Figure 4). This continental high air mass will hinder convection due to its stable lapse rate and relatively low moisture content. Once the ECSB forms, it will move rapidly inland and trigger convection but on a much weaker scale than Type I days. The WCSB will also set up, but it will remain relatively stationary since the synoptic wind flow will balance the on-shore flow. Once the two sea breezes meet on the western side of the peninsula,

enough forcing will result so that stronger convection will develop along the West Coast. This convection causes thunderstorms to form that are weaker than the Type I days and more short-lived, since the prevailing wind flow will push the storms over the Gulf of Mexico where they will decay (Blanchard and Lopez 1985).

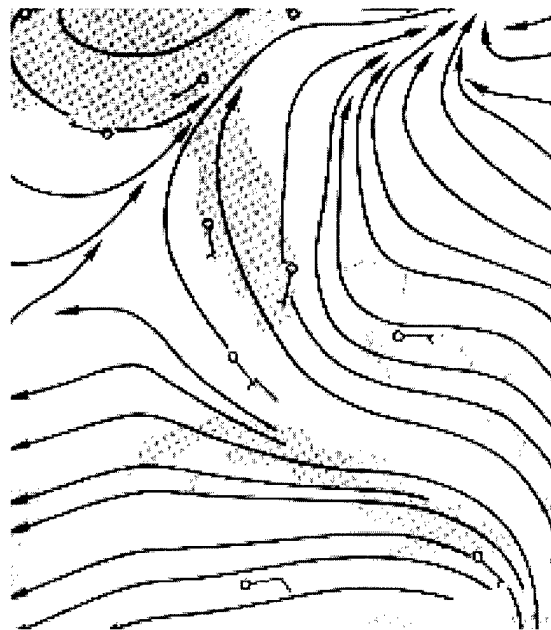


**Figure 4 Mean Synoptic Wind Field for Type II days (Blanchard and Lopez 1985)**

### **2.2.3 Type III days**

Type III days occur because of the Atlantic high which is now situated further east and south of the peninsula. The synoptic flow will be from the south-southwest (Figure 5) causing warm advection and corresponding vertical lifting to occur over the entire peninsula. Therefore, the atmosphere will become destabilized and conducive for convection. As expected, both sea breezes will set up along their corresponding coasts, but convection will start almost immediately. The synoptic wind flow will then push the

WCSB inland and keep the ECSB stationary. However, since this flow is from the south-southwest the WCSB will not reach the East Coast until late in the afternoon. Therefore, two lines of intense thunderstorms will set up: one along the East Coast and the other on the West Coast. They will steadily move eastward. Since the atmosphere is very unstable, the movement of the WCSB will cause convection that remains in the immediate area well after this sea breeze has moved further east. Unlike Type I and II days, Type III days exhibit an almost peninsula-wide echo area with most areas experiencing thunderstorms. This takes place because of the vertical velocities associated with the sea breeze circulations which are strongly modified by the synoptic scale forcing (Blanchard and Lopez 1985).



**Figure 5 Mean Synoptic Wind Field for Type III days (Blanchard and Lopez 1985)**

## **2.3 Sea Breeze Circulation Models**

The observed results have been simulated using various models. While these models do not take into account all meteorological parameters, they exhibit a good simulation of the atmosphere when a sea breeze circulation is present. Furthermore, the models can estimate the magnitude of the vertical velocities. These models give the meteorologist the ability to input different values for the synoptic wind flow so that the model can emulate the different types of days that occur over Florida. Nicholls *et al.* (1991) states that these models have shown conclusively that the convergence of the East and West Coast sea breezes are the primary controls on the timing and location of rapid convective development.

### **2.3.1 Model Run for Type 1 Days**

After the models ingested geostrophic wind values that match Type I days, the models produced excellent simulations of the actual sea breeze circulations. Once the two sea breezes had set up, the simulated ECSB moved inland at almost twice the speed of the WCSB. Before the two sea breezes converged, the vertical velocities were found to be on the order of 3 m/s. Once the sea breezes had merged, the vertical velocities grew to a maximum value of 8 m/s (Nicholls *et al.* 1991). One would expect strong thunderstorms to develop where this merging took place. This held true according to the actual observations. Another model simulation, using different parameters to simulate the sea breeze circulation, also found that with Type I geostrophic winds, the spatial distribution of thunderstorms matched closely with what was observed (Estoque 1962).

### **2.3.2 Model Run for Type II days**

The model runs also performed well when geostrophic winds matching those of the Type II day were introduced. As expected, convection developed along both coasts but the convection over the West Coast was advected over the ocean where it decayed. Eventually the sea breezes converged, but further west than was found in the Type I case. This occurred because the synoptic wind flow pushed the ECSB inland much more rapidly because the synoptic flow was almost normal to the sea breeze circulation (Estoque 1962). Once the two sea breezes merged, stronger convection occurred just inland of the West Coast. This convection formed thunderstorms that then moved over the Gulf and decayed while new cells continually developed along the sea breeze boundary. The vertical velocities determined by the models were approximately 6 m/s, which matched up with the findings that thunderstorms on Type II days were weaker than those of Type I days (Nicholls *et al.* 1991).

### **2.3.3 Model Run for Type III days**

As anticipated, the models performed well when using a southwesterly geostrophic wind component. Once the two sea breezes formed, convection occurred along both coasts. Then, as the WCSB moved eastward, convection spread quickly and a major portion of the peninsula became convectively active. When the two sea breezes merged, rapid thunderstorm development occurred approximately 10 km east of the center of the peninsula. These cells corresponded to a vertical velocity of approximately 6 – 8 m/s. There was one discrepancy between the model and the observed convection, however. This simulated convection did not last as long as the observations indicated. It was

determined that since the models only examined meso-scale features, the widespread destabilization of the atmosphere due to the synoptic forcing had not been considered in the model. Nicholls *et al.* (1991) believed that, indeed, synoptic-scale forcing mechanisms were responsible for this difference.

Many journal articles illustrate that the general synoptic wind flow is the major influence on where convection and thunderstorms will develop. By using a composite rainfall data set Blanchard and Lopez (1985) showed that three types of days exist. They also showed how sea breezes influence convection over the peninsula during the summer. Nicholls *et al.* (1990) and others, employing different techniques, used sea breeze models to show the relationship between sea breeze circulations and thunderstorm development. This information can be used to forecast thunderstorms over the peninsula and especially over the Cape. Given a specific synoptic wind field, meteorologists can determine the general development and movement of the sea breezes and where convection is most likely to form. A meteorologist would also be able to determine the intensity and duration of the thunderstorm(s) based on the strength of the sea breeze. Thus, a reliable forecast for thunderstorms over or near the Cape could be made several hours prior to a launch and operations could be tailored to protect personnel and assets.

## **2.4 Background Work**

A large amount of the background work for this thesis was accomplished by Charles Neumann in the 1960's. In this time period, he produced three technical reports: "Frequency and Duration of Thunderstorms at Cape Kennedy," "Frequency and Duration of Thunderstorms at Cape Kennedy Part II: Applications to Forecasting," and

“Thunderstorm Forecasting at Cape Kennedy, Florida: Utilizing Multiple Regression Techniques.” It should be noted that the name Cape Kennedy has since changed to Cape Canaveral. After he completed the first two technical notes, he wrote his third report that describes his forecasting technique and gives the current algorithm (NPTI) which is still being used to this day. This algorithm uses five predictors that are taken from a morning sounding and are used in producing a thunderstorm probability forecast and the timing of that thunderstorm. The following sections will discuss each article and then show how the algorithm works to produce these thunderstorm probabilities and timing.

#### **2.4.1 Frequency and Duration of Thunderstorms at Cape Kennedy Part I**

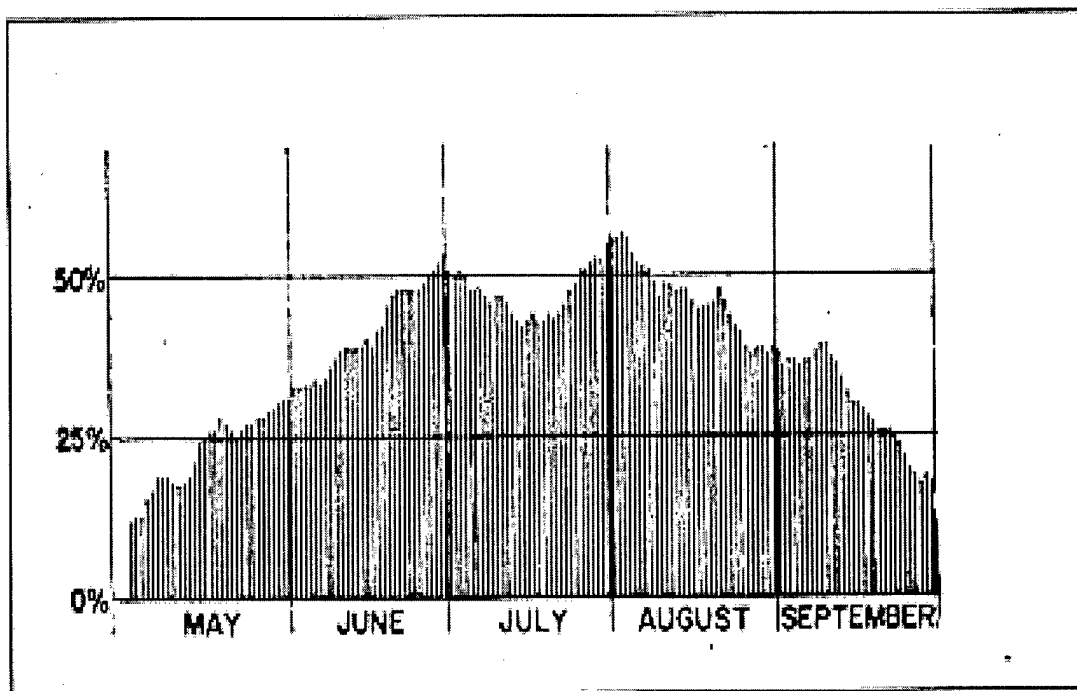
Neumann’s first report in 1968 examined summer (as described earlier) thunderstorm frequencies for Cape Canaveral for the period from 1950 – 1951 and 1957 – 1969 . This in-depth report examined conditional and nonconditional climatological probabilities of thunderstorms. Neumann found that a 15-day moving average best described this actual frequency (Neumann 1968). The equation used is shown in equation 1 and the plot can be seen in Figure 6.

$$A_n := \sum_{k=n-7}^{n+7} T_k \quad (1)$$

Where  $A_n$  = moving average on day number  $N$

$T_k$  = frequency of one or more thunderstorms on day  $k$

$N$  = total number of days over period of record



**FIGURE 6 NEUMANN 15-DAY MOVING AVERAGE USING 15 YEARS OF DATA (NEUMANN 1968)**

Neumann determined that the shape of this plot showed that thunderstorm probability was directly correlated to the day number (i.e., May 1 = 12.5% chance of thunderstorms occurring, Aug 1 = 52% chance of thunderstorms occurring). Neumann deduced the 15-day moving average by trial and error. He found that moving day averages other than 15 showed excessive smoothing or were too computationally expensive (Neumann 1968). This discovery led to the day number being a predictor variable in the NPTI.

#### **2.4.2 Frequency and Duration of Thunderstorms at Cape Kennedy, Part II**

Neumann's second study (1970) examined the winds associated with thunderstorms at Cape Canaveral. Specifically, he discussed the characteristics of the wind speed and direction of the 3,000 foot winds. The 3,000 foot winds were chosen because this wind level takes into account the sea-breeze effect on thunderstorm occurrence (Neumann 1970). He then plotted a series of ellipses of the wind and then split them into their  $u$  and  $v$  components. The same data was used as in his first study. The ellipses showed the magnitude of the  $u$  and  $v$  components and how they differ for each summer day.

Neumann used the regression estimation of event probabilities (REEP) method to bound the wind values. When using REEP, there is a slight chance that the forecast probabilities will be less than zero or greater than one (Wilks 1995). Thus, Neumann used the ellipses to bound the wind values to diminish the chance of an unrealistic probability result.

After Neumann had examined wind direction and wind speed separately, he examined them together as one parameter. He concluded that this combination of wind direction and speed was the most important factor for thunderstorm occurrence and should be included in his final algorithm (Neumann 1970).

#### **2.4.3 Thunderstorm Forecasting at Cape Kennedy, Florida, Utilizing Multiple Regression Techniques**

In Neumann's final report, he quickly reviewed his previous findings and discussed non-linear multiple regressions. He found that non-linear trends in the data were statistically significant and so they were included in the regression analyses (Neumann 1971). To account for this non-linear trend, he included 2<sup>nd</sup> or 3<sup>rd</sup> order polynomials to

represent the independent variables. It should be noted that Neumann had a pool of over 250 predictors, and after examining their correlations to thunderstorm occurrence, he determined that 5 predictors were the most important (Neumann 1971). The five predictors and respective polynomial functions can be found below:

$$F(X1) := A_0 + A_1 \cdot S + A_2 \cdot T + A_3 \cdot S \cdot T + A_4 \cdot S^2 + A_5 \cdot T^2 + A_6 \cdot S^3 + A_7 \cdot S^2 \cdot T + A_8 \cdot S \cdot T^2 + A_9 \cdot T^3 \quad (2)$$

$$F(X2) := B_0 + B_1 \cdot U + B_2 \cdot V + B_3 \cdot U \cdot V + B_4 \cdot U^2 + B_5 \cdot V^2 + B_6 \cdot U^3 + B_7 \cdot U \cdot V + B_8 \cdot U \cdot V^2 + B_9 \cdot V^3 \quad (3)$$

$$F(X3) := C_0 + C_1 \cdot RH + C_2 \cdot RH^2 + C_3 \cdot RH^3 \quad (4)$$

$$F(X4) := D_0 + D_1 \cdot SSI + D_2 \cdot SSI^2 \quad (5)$$

$$F(X5) := E_0 + E_1 \cdot DAY + E_2 \cdot DAY^2 \quad (6)$$

Where S & T = *u* and *v* components of 850 mb wind in knots

U & V = *u* and *v* components of 500 mb wind in knots

RH = 600-800 mb mean relative humidity in percent

SSI = Showalter Stability Index in degrees Celsius

DAY = Day number

X1 = 850 mb wind in knots

X2 = 500 mb wind in knots

X3 = 600-800mb mean relative humidity in percent

X4 = Showalter Stability Index in Degrees Celsius

X5 = Day number

Neumann set thunderstorm occurrence to a value of zero which corresponded to no thunderstorm occurring and the value one to a thunderstorm occurring. For instance, using Equation 2, if a thunderstorm occurred on a given day, a one was placed in F(X1) and the corresponding  $u$  and  $v$  components for the day were inserted into the right side of the equation. This was accomplished for all thunderstorm and no thunderstorm days. The resulting equations were then linearly regressed and the coefficients were found. The resulting F(X) values were then inserted into the prediction equations given below:

$$P(\text{MAY}) := H_0 + H_1 \cdot F(X1) + H_2 \cdot F(X2) + H_3 \cdot F(X3) + H_4 \cdot F(X4) + H_5 \cdot F(X5) \quad (7)$$

$$P(\text{JUN}) := I_0 + I_1 \cdot F(X1) + I_2 \cdot F(X2) + I_3 \cdot F(X3) + I_4 \cdot F(X4) + I_5 \cdot F(X5) \quad (8)$$

$$P(\text{JUL}) := J_0 + J_1 \cdot F(X1) + J_2 \cdot F(X2) + J_3 \cdot F(X3) + J_4 \cdot F(X4) + J_5 \cdot F(X5) \quad (9)$$

$$P(\text{AUG}) := K_0 + K_1 \cdot F(X1) + K_2 \cdot F(X2) + K_3 \cdot F(X3) + K_4 \cdot F(X4) + K_5 \cdot F(X5) \quad (10)$$

$$P(\text{SEP}) := L_0 + L_1 \cdot F(X1) + L_2 \cdot F(X2) + L_3 \cdot F(X3) + L_4 \cdot F(X4) + L_5 \cdot F(X5) \quad (11)$$

Where X1 = 850 mb wind in knots

X2 = 500 mb wind in knots

X3 = 600-800 mb mean relative humidity in percent

X4 = Showalter Stability Index in degrees Celsius

X5 = Day number

Once all  $F(X)$ 's were known, Neumann used the distribution of thunderstorm occurrence again and set this value equal to the right side of the equations above. After being linearly regressed, the constants in the prediction equations (7) – (11) were found. In order to find the probability of thunderstorm occurrence, all that was needed was the 850 mb and 500 mb wind components, mean relative humidity, Showalter Stability Index and day number. These values were used in conjunction with the already known coefficient values and a probability was produced. The coefficient values for equations (2) – (11) can be found in Appendix A.

#### **2.4.4 Thunderstorm Starting Time**

To determine the thunderstorm starting time, Neumann calculated the average starting time of thunderstorms using his data set as described earlier. He found the sample mean to be 1434 Eastern Standard Time (EST). Neumann then assumed these times were normally distributed about the mean and that two-thirds of the thunderstorm starting times ( $\pm 1$  standard deviation) would be expected between the hours 1204 and 1705 EST (Neumann 1971). According to Kachigan (1991), for a given population there may be any number of different parameters in which a person is interested, e.g., mean, median, standard deviation. One approach to find the estimated parameter of interest is to obtain a single value based upon a sample of observations, a value which is thought to be the best possible approximation of the true value of the population parameter (Kachigan 1991). What Neumann found was that when looking at four parameters, he could produce an estimate of the sample standard deviation as given above. Neumann determined that by using the 850 mb orthogonal winds, probability of thunderstorm

occurrence and day number, the estimate of standard deviation produced was within the actual sample standard deviation from his data set (+/- 1 ½ hours). To find the estimated thunderstorm start times, a third-order polynomial expansion of the four parameters was performed. This equation can be found in Equation 12

$$\begin{aligned} \text{Thunderstorm Start Time} := & C_1 + C_2 \cdot y + C_3 \cdot y^2 + C_4 \cdot y^3 + C_5 \cdot x + C_6 \cdot x \cdot y + C_7 \cdot x \cdot y^2 + C_8 \cdot x^2 + C_9 \cdot x^2 \cdot y \dots \\ & + C_{10} \cdot x^3 + C_{11} \cdot w + C_{12} \cdot w \cdot y + C_{13} \cdot w \cdot y^2 + C_{14} \cdot w \cdot x + C_{15} \cdot w \cdot x \cdot y + C_{16} \cdot w \cdot x^2 \dots \\ & + C_{17} \cdot w^2 + C_{18} \cdot w^2 \cdot x + C_{19} \cdot w^2 \cdot y + C_{20} \cdot w^3 + C_{21} \cdot v + C_{22} \cdot v \cdot y + C_{23} \cdot v \cdot y^2 \dots \\ & + C_{24} \cdot v \cdot x + C_{25} \cdot v \cdot x \cdot y + C_{26} \cdot v \cdot x^2 + C_{27} \cdot v \cdot w + C_{28} \cdot v \cdot w \cdot y + C_{29} \cdot v \cdot w \cdot x \dots \\ & + C_{30} \cdot v \cdot w^2 + C_{31} \cdot v^2 + C_{32} \cdot (v^2 \cdot y) + C_{33} \cdot v^2 \cdot x + C_{34} \cdot v^2 \cdot w + C_{35} \cdot v^3 \end{aligned}$$

Where  $y$  = Thunderstorm Probability

$x$  = Day Number

$v$  = 850 mb  $u$  wind component in knots

$w$  = 850 mb  $v$  wind component in knots

In order to solve this equation, Neumann took his data set and simultaneously solved Equation 12 using the starting times of actual thunderstorms. On completion, 35 time coefficients were produced and used in the final algorithm. These constants can be found in Appendix B.

Finally, Neumann mentioned his verification process. He used the month of June as a verification month and applied the equations to arrive at thunderstorm probabilities and thunderstorm starting time. After completing this process, he determined that his thunderstorm starting times had an estimated margin of error of +/- 1 ½ hours of the forecast starting time (Neumann 1971).

### 3. Methodology

This chapter discusses the data used in this thesis. It is important to understand how the variables were calculated and what tools were used to account for missing data. In addition, the methods used to recreate the NPTI as well as the new algorithms are discussed.

#### 3.1 Data Used

Surface observations and upper air data from 1950 – 1998 recorded at the official observation site at Cape Canaveral was used. This data was obtained from AFCCC and was given in Microsoft® Excel spreadsheets. Table 1 shows which years had surface observations available for analysis. Table 2 shows which upper air data was available for analysis.

**Table 1 Available Surface Observations**

	May	June	July	August	September
Years with	1950 - 1977	1950 - 1977	1950 - 1977	1950 - 1977	1950 - 1977
Observations	1987 - 1998	1987 - 1998	1987 - 1998	1987 - 1998	1987 - 1998

**Table 2 Available Upper Air Observations**

	May	June	July	August	September
Years	1950 - 1970	1950 - 1969	1950 - 1969	1950 - 1969	1950 - 1969
Available	1983 - 1998	1983 - 1998	1983 - 1998	1983 - 1998	1983 - 1998

These surface observations were then matched up with the upper air data. Microsoft<sup>®</sup> Access was used to perform this task. The day and year of each surface observation was compared to the day and year of each upper air observation and when a match occurred, this upper air observation was placed in a new spreadsheet. When this was completed, all surface observations had corresponding upper air observations.

One important stipulation given by the Patrick AFB forecasters was that the forecast needed to be produced by 1100UTC. Therefore, any upper air observations taken after this time would be of no use to the forecaster for that day. Unfortunately, on some thunderstorm days, upper air observations were only taken after 1100UTC. To screen the data, Microsoft<sup>®</sup> Access was used and both surface observations and upper air data were examined. If the time for any given day's upper air data was after 1100UTC, then the data was removed together with the matching surface observation. The resulting spreadsheet had surface observations for thunderstorm days with upper air data that was received before 1100UTC only. Table 3 below shows all years that had surface observations and upper air data which had been received before 1100UTC.

**Table 3 Years with Available Surface and Upper Air Observations**

	May	June	July	August	September
	1950 - 1969	1950 - 1969	1950 - 1969	1950 - 1969	1950 - 1953
Years	1987 - 1988	1987 - 1988	1987 - 1988	1987 - 1988	1958 - 1969
Available	1992 - 1998	1991 - 1996	1992 - 1998	1992 - 1998	1987 - 1988
		1998			1998

### 3.2 Interpolation

The upper air data was examined for any values of “999.” Values of “999” attributed to a parameter indicated that that parameter was missing. Another interesting note is that prior to 1970, all upper air observations were reported from 1000 mb upwards in 50 mb increments. After 1970, all pressure levels recorded by the rawinsonde were reported. To recreate the NPTI it was decided that using 50 mb increments was the most prudent method of using data. For this reason, one interpolation technique was used but the methods were different for the different types of data given.

To find missing data from 1950 – 1970, an Interactive Data Language<sup>®</sup> (IDL) program was created. The method used to interpolate is the same method a meteorologist would use when given a Skew – T diagram with missing data. When plotting a Skew – T diagram with missing values, the meteorologist will place an X at the pressure level where this missing parameter is located. The meteorologist then draws a line from value to value. When an X is present the meteorologist will connect the lower and upper value, drawing a straight line through the pressure level where the missing value is located. This interpolated line now gives an estimate of the missing value. The interpolation scheme created does this in exactly the same way. This is scientifically sound, assuming the missing parameters do not change drastically over a small vertical distance. The most common parameters missing were the temperature and dewpoint. This program can be found in Appendix C.

For the data from 1971 and onwards, a Microsoft<sup>®</sup> Qbasic program was written. This program was written to ensure that all upper air data started at 1000 mb and increased upwards in 50 mb increments. The program determined if a 50 mb level was missing. If

so, the level above and below was extracted and the missing pressure level values was exported to a new file. Once completed all upper air data was reported from 1000 mb to 500 mb in 50 mb increments. This Quick Basic program can be found in Appendix D.

### 3.3 Computations

Before being able to compute a probability and time for thunderstorm occurrence, two computations must be performed to arrive at the predictors needed. Neumann determined that the 800 – 600 mb mean relative humidity was important as well as the Showalter Stability Index (SSI). These computations are described in the next two sections.

#### 3.3.1 Mean Relative Humidity

In Neumann's reports, he determined that the 800 – 600 mb mean relative humidity gave a reasonable approximation for the amount of moisture in the atmosphere (Neumann 1971). He also showed that the correlation between moisture and thunderstorm occurrence was high and therefore should be included in his algorithm. The equation used to find the mean relative humidity in this thesis is adapted from the Air Weather Service's (AWS) Technical Report 83/001 (Duffield and Nastrom 1983). The equation used is as follows:

$$\text{MeanRH} := \frac{1}{\ln(800) - \ln(600)} \cdot \sum_{i=1}^4 \frac{((\text{RH}(i) + \text{RH}(i+1)) \cdot (\ln(P(i)) - \ln(P(i+1))))}{2} \quad (13)$$

Where: RH(i) = relative humidity at 800 mb at RH(1), 750 mb at RH(2), etc.  
P(i) = pressure at level i so P(1) = 800 mb, P(2) = 750 mb, etc.

The IDL<sup>®</sup> program used to compute the mean relative humidity can be found in Appendix E.

### 3.3.2 Showalter Stability Index

The SSI is a thunderstorm index that is used to determine the possibility and severity of a thunderstorm occurrence. The process to find the SSI manually is lengthy when dealing with a large data set. Once again, an IDL<sup>®</sup> program was adapted from the AWS Technical Report 83/001 (Duffield and Nastrom 1983). The description of the values generated by using the SSI are given in Table 4. The IDL program that calculates the SSI can found in Appendix F.

**Table 4 Values of SSI and Descriptions**

Value of SSI	Description
1 to 3	Thunderstorms Possible
0 to -3	Unstable Thunderstorms Probable
-4 to -6	Very Unstable Good Heavy Thunderstorm Potential
< -6	Extremely Unstable Good Strong Thunderstorm Potential

### 3.4 Types of Regression

Regression is that part of statistics which deals with the investigation of the relationship between two or more variables related in a non-deterministic fashion (Devore 1995). There are many different types of regressions, and the following sections

explain the two used in this thesis: Regression Estimation of Event Probabilities (REEP) and logistic regression.

### **3.4.1 REEP**

REEP is a regression approach that, in this case, estimates thunderstorm occurrence. Neumann used this method because it involves only multiple linear regression to derive a forecast equation for thunderstorm occurrence. Theoretically, it is possible to forecast probabilities that are either negative or greater than one. That is, if a probability of .10 is produced there is a 10% chance of a thunderstorm occurring while a probability of -.27 is rounded to a 0% chance of a thunderstorm occurring.

Neumann applied REEP but also took into account the non-linear response of the predictor values. By using polynomial equations to describe each predictor, this non-linear response is accounted for. Each polynomial is then linearly regressed against thunderstorm occurrence. Once accomplished, a set of coefficients is produced for each polynomial function. Now a value for the predictors needed in the second regression can be calculated given the coefficients and the inputted predictor values.

These values are then passed into a second regression. Once again, the new predictors are set equal to thunderstorm occurrence and linearly regressed. As before, new coefficients are created and used find the probability of the occurrence. In order to find the probability of a thunderstorm occurring, the predictors are inserted into the first linear equation. The computed predictor values are then placed into the second regression and a final probability is given.

To find the timing coefficients, Neumann used simple linear regression. He realized that the non-linear trends in the predictors also needed to be accounted for in this equation. The resulting equation produced 35 unknown coefficients. By setting this equation equal to thunderstorm starting time, he simultaneously solved the equations by using linear regression and determined the timing coefficients. Similarly, to find the starting time it was necessary to insert the predictor values into the final equation and a thunderstorm starting time was produced.

### 3.4.2 Logistic Regression

A more theoretically satisfying regression method is a technique called logistic regression (Wilks 1995). Neter (1983) found that when the predictand is not continuous, a curvilinear function should be used. Logistic regression accomplishes this by assuming an exponential distribution. This causes the regression to be bounded by zero and one, thus preventing the possibility of an unrealistic probability. The equation used in logistic regression is as follows:

$$E(Y) := \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}} \quad (14)$$

Where  $\beta_0$  = coefficient that changes the horizontal placement of the curve

$\beta_1$  = coefficient that changes the slope of the curve

### 3.5 Research Approach

Once all data was interpolated and calculations performed, new algorithms were developed. The following sections discuss how the algorithms were created.

#### 3.5.1 Neumann Probability (NP) and Neumann Timing (NT)

The NPTI was recreated and used as a baseline from which to test the new methods described in this thesis. Mathcad<sup>®</sup> was used to process the algorithms. Within the program, the first calculation needed was to decompose the 850 mb and 500 mb winds into their corresponding orthogonal components. The formulas used are shown below:

$$U := \sin \left[ \left( \text{Dir} \right) \cdot \left( \frac{\pi}{180} \right) + \pi \right] \cdot \text{SPD} \quad (15)$$

$$V := \cos \left[ \left( \text{Dir} \right) \cdot \left( \frac{\pi}{180} \right) + \pi \right] \cdot \text{SPD} \quad (16)$$

Where  $U = u$  component for 850 or 500 mb wind

$V = v$  component for 850 or 500 mb wind

Dir = wind direction at either 850 or 500 mb in degrees

Spd = wind speed at either 850 or 500 mb in knots

Day number was calculated by first examining the month. If the month examined was May, then it was known that 150 days had already passed. To find the day number for a date in May that date was added to 150. This same method was repeated for the other

months. The mean relative humidity and SSI were extracted using Mathcad's<sup>®</sup> built-in functions.

To perform the first regression, a matrix of coefficients was entered into Mathcad<sup>®</sup>. In this first case, these were the same coefficients that Neumann used. Then each predictor was inserted into the equation and solved. These values were then passed to the second regression equation, with corresponding coefficients, and a probability for thunderstorm occurrence was produced.

This probability, along with the 850 mb  $u$  and  $v$  components and day number, were then used to find the thunderstorm starting time. Neumann's timing coefficients were entered into Mathcad<sup>®</sup> and the timing equation was solved. The resulting number is expressed in hours and fractions of hours. Next it had to be expressed in hours and minutes. A simple program was used to convert the given number into hours and minutes. Here, the minutes were represented as a proportion of an hour so they were multiplied by 60 to express the number of minutes in real time. The entire program can be found in Appendix G.

### **3.5.2 Logistic Regression Probability (LRP) and Neumann Timing (NT)**

In this thesis, the first algorithm created used logistic regression to find thunderstorm probability, which was passed into Neumann's timing scheme using his coefficients to produce a starting time. Everitt (1999) found that using logistic regression provided a more accurate forecast probability than linear regression did. The logistically regressed coefficients calculated by Everitt are listed in Appendix H.

Mathcad<sup>®</sup> was again used to find the probability and timing of thunderstorms. Everitt's (1999) previously calculated logistic coefficients were used in the regression. Of important note is that the Everitt's (1999) logistical coefficients given for May produced incorrect probabilities. The resulting probabilities varied from a value of 10 up to 100. Since all other months produced reasonable probabilities between 0 and 1, May was dropped as a month to be examined. One possible reason for this aberration was a typographical error in Everitt's thesis.

The new logistic probability of thunderstorm occurrence was calculated and was placed in Neumann's timing scheme. The same method as mentioned earlier was used to calculate the thunderstorm starting time. This program can found in Appendix I.

### **3.5.3 NP and Linearly Regressed Timing Coefficients (LRTC)**

The second algorithm created in this thesis uses Neumann's method of finding the probability of thunderstorm but employs new linearly regressed timing coefficients. In this method a new linear regression was executed to find new timing coefficients. This was accomplished by first calculating the predictor values for each combination given in Equation 12. The corresponding thunderstorm starting time for that day's data was then set equal to the calculated predictor values. After linear regression, 35 new timing coefficients were created. These new timing coefficients can be seen in Appendix J.

To find this new starting time, Neumann's probability of thunderstorm occurrence was placed in the timing scheme using the newly derived timing coefficients and new thunderstorm starting times were produced. This program can be found in Appendix K.

#### **3.5.4 Logistic Regression Probability (LRP) and Linearly Regressed Timing Coefficients (LRTC)**

The final algorithm used both logistic regression to find thunderstorm probability and the timing scheme using new linearly regressed coefficients. The new linearly regressed coefficients using logistically regressed probabilities can be found in Appendix L. This was very easy to accomplish since both of these steps had been previously performed. The calculated logistic probabilities were inserted into the timing scheme with the new timing coefficients. As before, another thunderstorm starting time was produced. This program can be found in Appendix M.

#### **3.5.5 Average Thunderstorm Starting Time**

One last method used to examine thunderstorm start times is to use the average thunderstorm start time calculated from the entire data set. All thunderstorm start times were placed in one spreadsheet and then an average of all these times was produced. This average start time can then be compared to actual thunderstorm start times. The calculated average thunderstorm start time was found to be 1504 EST.

#### **3.6 Verification Data Set**

In order to verify the new timing schemes, some of the original data was withheld from the regressions and used to calculate thunderstorm starting times. These calculated thunderstorm forecast starting times were used to compare the accuracy to the actual thunderstorm starting times. Sufficient data was available that 20% could be withheld for verification purposes. Twenty percent of the data was randomly removed using

Mathcad<sup>®</sup>. First, Mathcad<sup>®</sup> randomly picked row numbers. These extracted values were stacked into a new matrix. The day number and years were used to remove the same day number and year manually in the regression equations. This ensured that the data in the verification set would be independent of the new regressions. These extracted rows were used in each method for finding thunderstorm probability and starting time. Because there were five ways to examine each month (NPTI, LRP with NT, NT with LRTC, LRP with LRTC, average thunderstorm start time) it was imperative that the same random rows were extracted for each type of regression in each month. Fortunately, Mathcad's<sup>®</sup> seed function ensured that when using the same data set to pull random data samples, the same random numbers would be extracted. For instance, when using Neumann's method to find thunderstorm starting time a certain number of rows were randomly removed from the month of June for verification purposes. These same rows in June were removed from the other four methods which ensured that the verification data set was the same for each method. This program can be found in Appendix N.

#### 4. Statistical Analysis and Results

Statistical analysis is an important aspect of this thesis. These analyses quantify the improvement achieved and identify which timing scheme should be implemented for each month. Descriptive statistics are the best way to examine how these schemes perform. They examine the relationships between the actual thunderstorm starting time and the forecast starting times. For each month, five methods are examined: NP with NT, LRP with NT, NP with LRTC, LRP with LRTC, and finally average thunderstorm start time. Once all the different times were produced they were subtracted from the actual thunderstorm start time. If the forecast time was later than the actual time the resultant was less than zero. If the forecast time was earlier than the actual time the resultant was greater than zero. By using this method, the estimated mean and standard deviations for each method could be examined. Of important note is that the sponsor for this thesis was more concerned with achieving the smallest standard deviation possible. Therefore, when looking for the “best” method, the smallest standard deviation will be considered. The next sections will discuss the results by month.

##### 4.1 June Timing Schemes

For the month of June, 65 days were used as the test set. The starting times were produced as mentioned in Chapter 3 and then compared to the actual starting time. The average mean error and standard deviation can be seen in Table 5.

**Table 5 Mean and Standard Deviation of Error between Actual Starting Time and Forecast Starting Time for June**

	NP with NT	LRP with NT	NP with LRTC	LRP with LRTC	Avg Tstorm Time
MEAN	- 44 min	- 46 min	-1 hr	-1 hr 13 min	- 24 min
STD DEV	3 hr 7 min	2 hr 49 min	2 hr 54 min	4 hr 9 min	2 hr 13 min

It can be seen from Table 5 that the smallest mean error of thunderstorm starting time is achieved by using the average thunderstorm start time. It should be noted that when using NP with NT, LRP with NT and the method using NP with LRTC all produce standard deviations that differ on the order of minutes. Unfortunately, any forecasts with standard deviations as large as those in Table 5 are not an asset for a forecaster. For example, if while using NP with NT a forecast starting time of 1200 EST is produced, then thunderstorms can be expected between 0853 EST and 1507 EST. Obviously, this range is too wide to be very useful operationally.

Another way to examine this data is to plot a normal distribution with the means and standard deviations given in Table 5. This normal plot will give a visual idea as to how the methods differ. In order to plot the graph, it must first have a known mean ( $\mu$ ) and standard deviation ( $\sigma$ ). These values are then inserted into the equation given below:

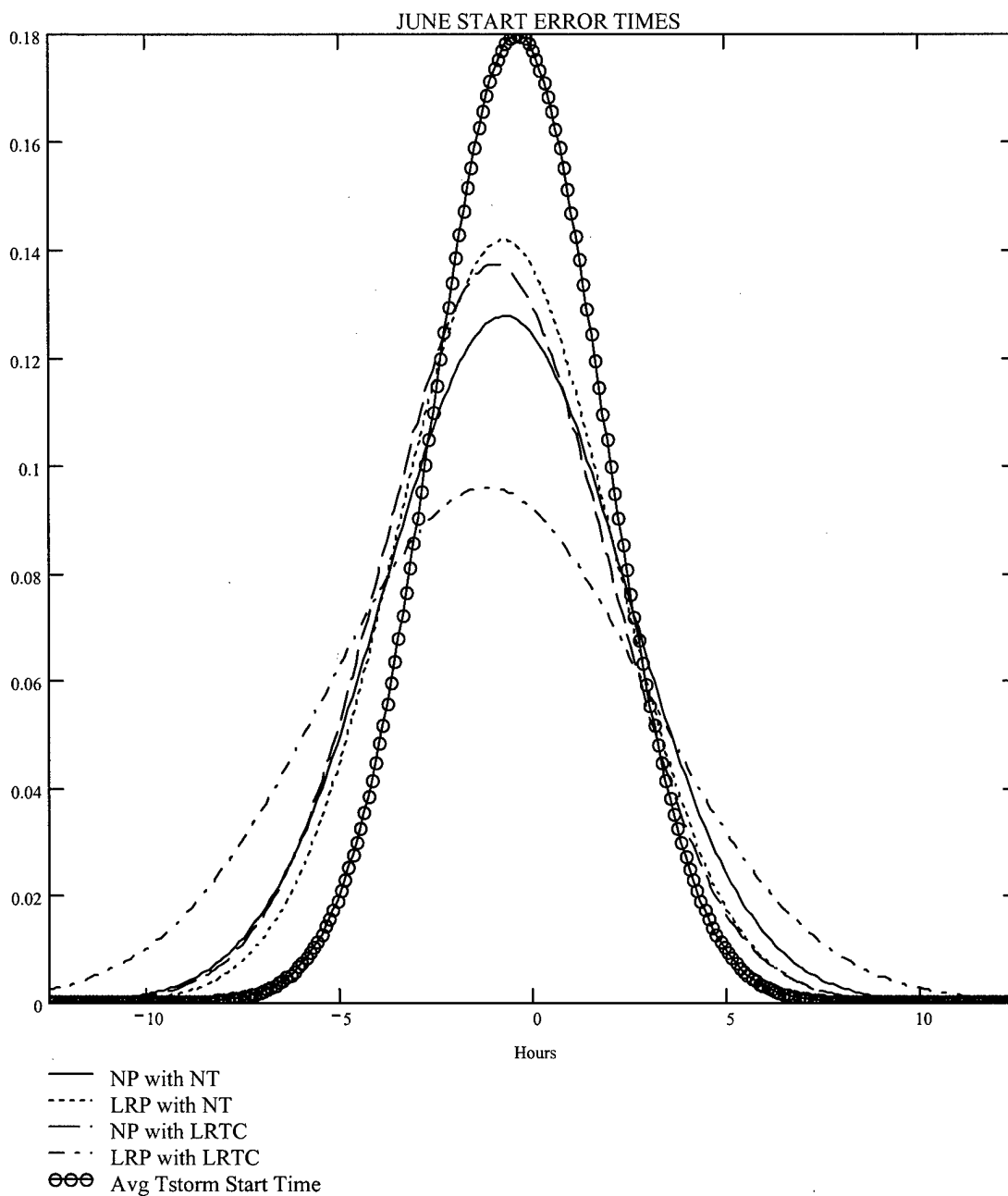
$$f(x; \mu, \sigma) = \frac{1}{\sqrt{2 \cdot \pi} \cdot \sigma} \cdot e^{\frac{-(x - \mu)^2}{2 \cdot \sigma^2}} \quad (17)$$

Equation 17 is an example of a probability density function (PDF). PDF's are the continuous, theoretical, analogs of the familiar histogram and must satisfy Equation 18 (Wilks 1995).

$$\int f(x) dx = 1 \quad (18)$$

No specific limits of integration have been included in Equation 18 because different probability densities are defined over different ranges of the variable  $X$ . Unfortunately, the height of  $f(X)$ , when evaluated at a particular value of  $X$ , is not meaningful in a probability sense. This happens because the probability is proportional to the area under the curve, and not to the height. For example, if  $X=1$ , by itself  $f(1)$  is not meaningful in terms of the probability of  $X$  since the probability of  $X=1$  is infinitesimally small (Wilks 1995). It is significant, however, to find values surrounding  $X=1$  (say from  $X=.95$  to  $X=1.05$ ). This is accomplished by integrating Equation 17 from .95 to 1.05.

Figure 7 below is a plot of the PDF for the five different methods used to find thunderstorm forecast start time. In order to plot the graph, all that is needed are values of  $X$ . These are arbitrarily selected so that the graph encompasses all values where  $y$  is positive. By plotting the five means and standard deviations in this way, one plot can visually show how the means and standard deviations differ for each method. The values on the  $y$  axis are simply values of the function given in Equation 17 for different  $X$  values. It must be remembered that the values on the  $y$  axis are not probabilities for the reasons mentioned above. The probabilities are equivalent to the area under the curve for given values of  $X$  (in this case, hours) and are found by integrating equation 17 for given  $X$  values.

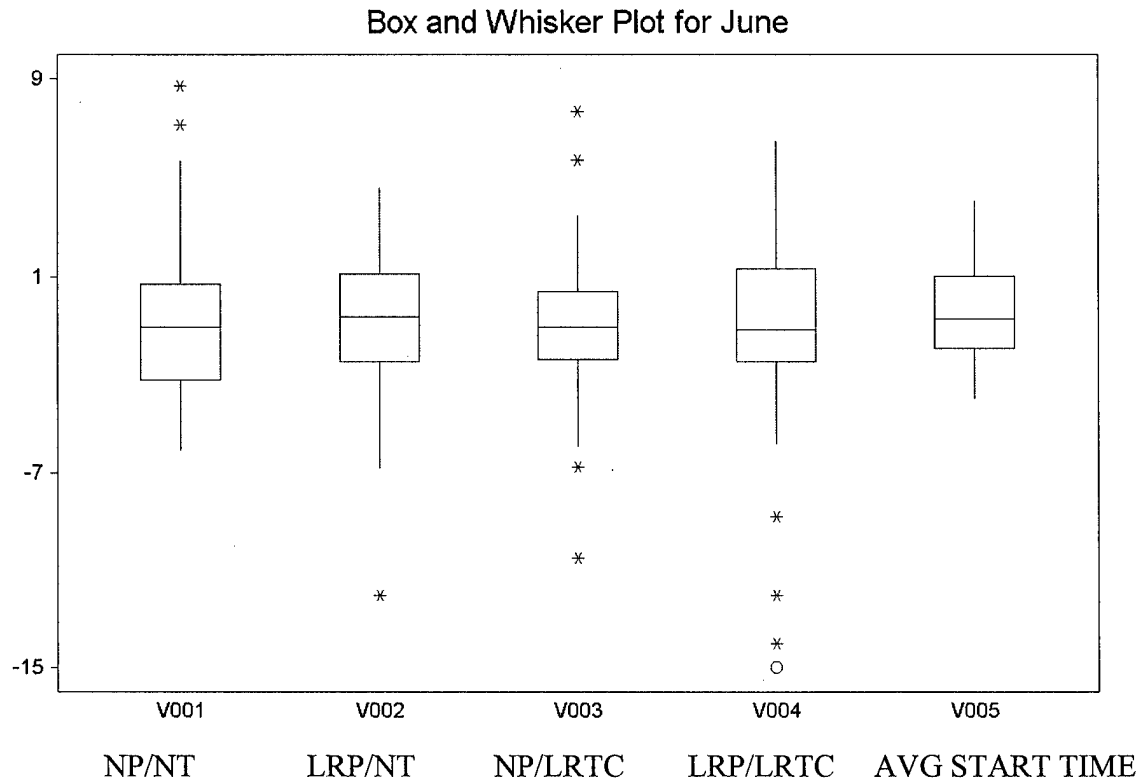


**Figure 7 Graph of Mean and Standard Deviation Error from Actual Starting Time for June**

From this figure, it can be seen that using logistical regression to find the probability of thunderstorm occurrence whilst using the new linearly regressed timing coefficients (LRP

with LRTC) produces the worst result. The standard deviation for the LRP with LRTC causes the normal curve to be more widely dispersed about the mean than the other four methods. Obviously then, using the average thunderstorm time to forecast actual start times is the method to use since the standard deviation is much smaller as shown by the tightness of the curve around the mean.

The final method to examine these results is to produce a Box and Whiskers plot using the computer program, Statistix<sup>®</sup>. Each box plot in Figure 8 is composed of a box and two whiskers. The box encloses 50% of the data. This box is also bisected by a line which represents the value of the median. The vertical lines at the top and bottom of the box are called the whiskers, and they indicate the range of “normal” data values. Whiskers always end at the value of an actual data point and cannot be longer than  $1\frac{1}{2}$  times the size of the box. Extreme values are displayed as \* for possible outliers and O for probable outliers. Possible outliers are values that are outside the box boundaries by more than  $1\frac{1}{2}$  times the size of the box. Probable outliers are values that are outside the box boundaries by more than 3 times the size of the box. One important use of the box and whiskers plot is the ability to graphically compare several batches of data at one time (Wilks 1995).



**Figure 8 Box and Whiskers Plot of Error Data for June**

Figure 8 confirms that LRP with LRTC produces the worst result. This is shown by the large number of outliers on the plot. LRP with LRTC has the largest number of outliers of all methods. In addition, all the outliers are in the negative region of the plot which shows this method has a tendency to forecast a thunderstorm start time many hours earlier than when the thunderstorm really occurred. Using the average thunderstorm start time is the best method to use because this plot shows these forecast start times are reasonably symmetric about the median, the whiskers are smaller indicating the forecast times will be closer to the actual time of thunderstorm occurrence and there are no outliers.

After examining all previous figures, it is apparent that the average thunderstorm start time method should be used when forecasting thunderstorm start times for the month of June. The average thunderstorm start time has both the smallest mean error and standard deviation from the actual thunderstorm start time. This ensures that the spread of estimated thunderstorm starting times will be closer to the actual time. The normal plot visually demonstrates how the average thunderstorm time creates a tighter spread of start time. The box and whiskers plot also showed that when using the average thunderstorm start time to predict actual start times it produces a closer forecast start time than the other methods examined.

#### 4.2 July Timing Schemes

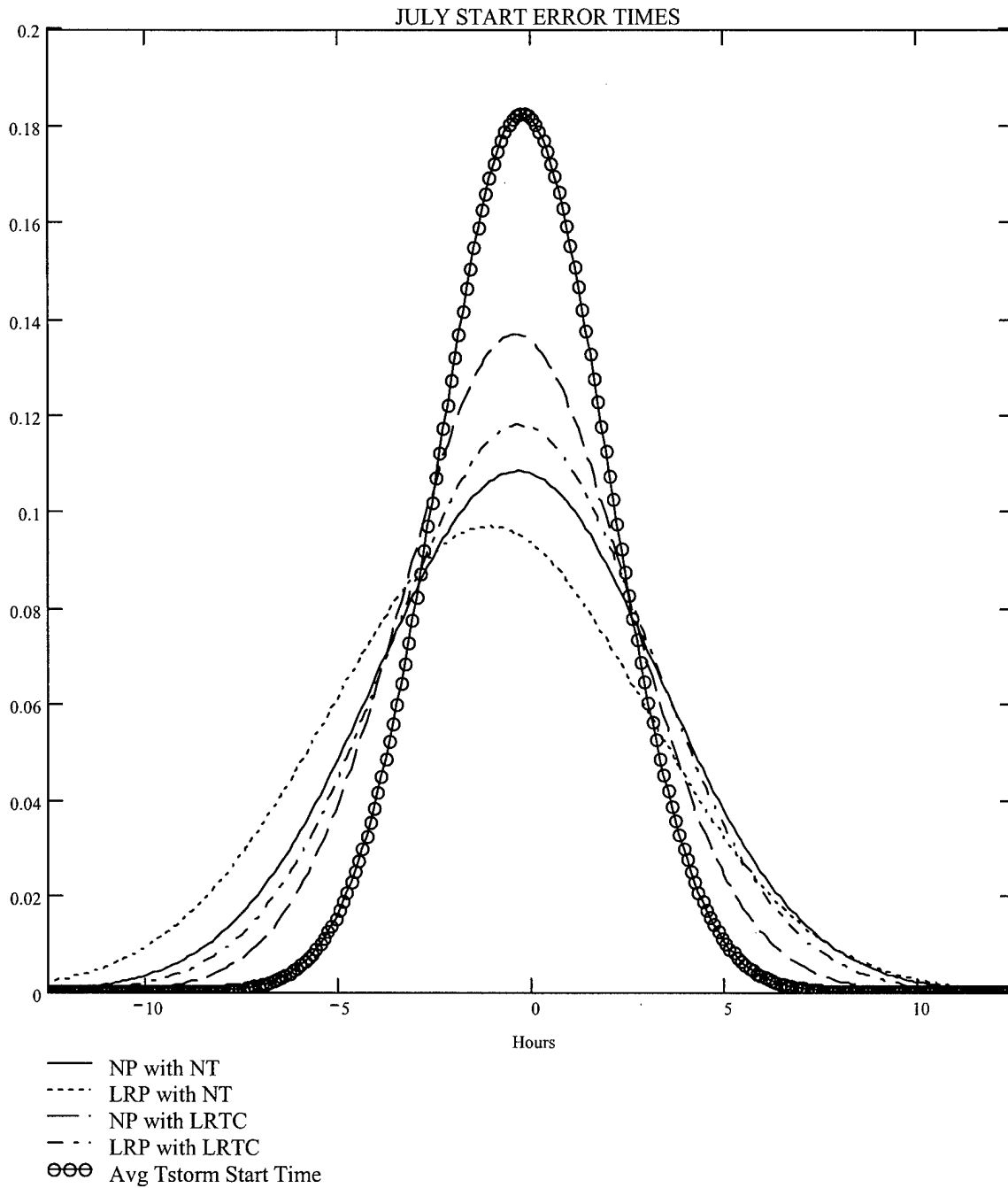
For the month of July, 86 days were used as the verification set. The starting times were produced and then compared to the actual starting time. The average mean error and standard deviation can be found in Table 6.

**Table 6 Mean and Standard Deviation of Error between Actual Starting Time and Forecast Starting Time for July**

	NP with NT	LRP with NT	NP with LRTC	LRP with LRTC	Avg Tstorm Time
MEAN	- 22 min	-1 hr 7 min	-26 min	-19 min	-14 min
STD DEV	3 hr 41 min	4 hr 6 min	2 hr 55 min	3 hr 23 min	2 hr 11 min

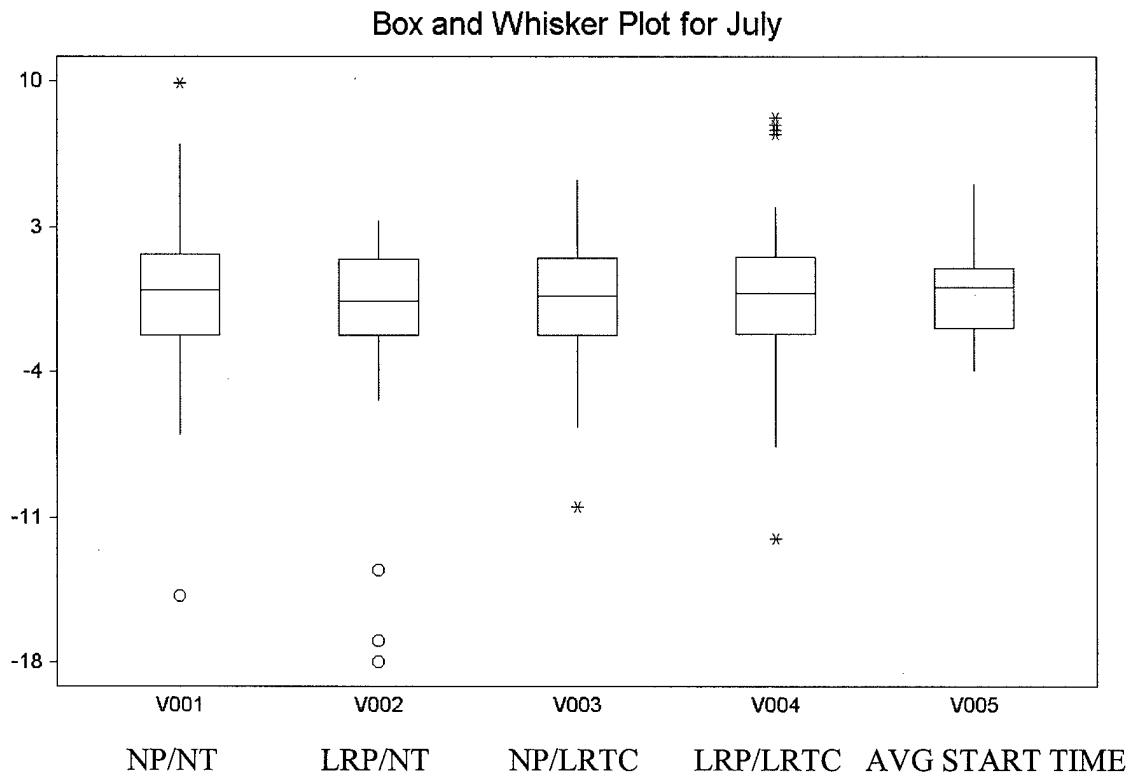
Using the average thunderstorm start time produces the best timing standard deviation outperforming all other methods by a minimum of 44 minutes. The mean starting error

is also the best since it only misses on average by 14 minutes and so would appear to be the best method to use. Using the same methods as described in section 4.1, this information can be normalized and plotted. This graph can be seen in Fig 9.



**Figure 9 Graph of Mean and Standard Deviation Error from Actual Starting Time for July**

From this figure, it is easily seen that using logistical regression to find the probability and using Neumann's timing coefficients (LRP with NT) produces the worst result because the peak is the lowest of all methods plotted indicating a large standard deviation. It is also apparent that using average thunderstorm starting time is the best method to use since the peak is much higher indicating a tighter spread around the mean. The box and whisker chart for this data set can be seen in Figure 10.



**Figure 10 Box and Whiskers Plot of Error Data for July**

Upon examination of Figure 10 it is not so readily apparent which method achieves the best results. All box sizes are relatively the same size so 50% of the data for each method is within the same error margin. The whiskers for each plot differ greatly, however. The

smallest whiskers occur when LRP with NT is used and the next smallest occur when the average thunderstorm starting time is used. From this graph alone, it appears that LRP with NT would be the best method to use. The 3 probable outliers which occur when using this method, however, are much worse than when using the average thunderstorm start time. Since this thesis is concerned with achieving the “best” start time, the average thunderstorm start time appears to be the best method to use.

The average thunderstorm start time should be used when forecasting thunderstorm times for July. Not only does it have a smaller standard deviation to the other methods examined, the start time it does produce has a higher chance of being closer to the actual time than the other methods.

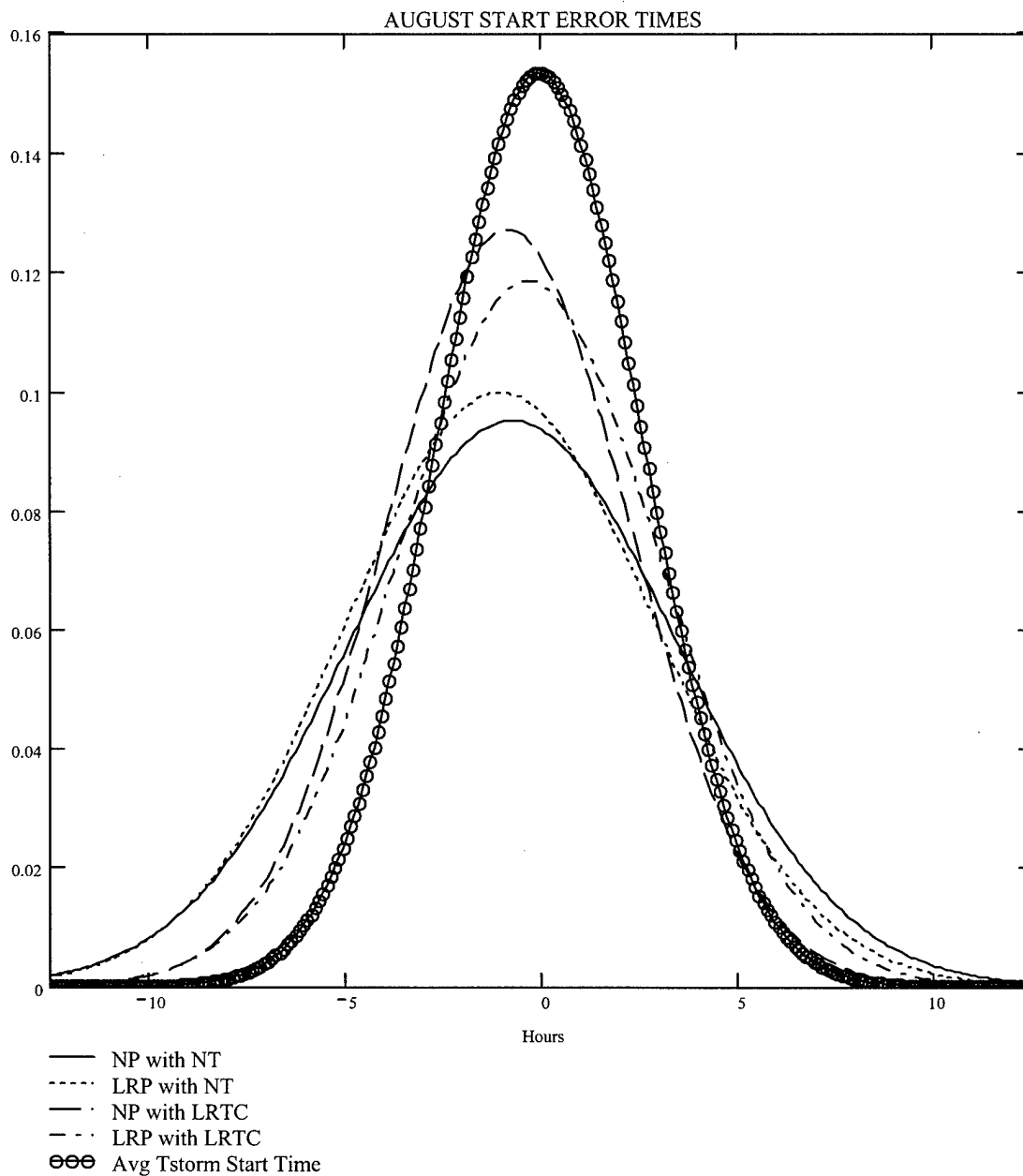
### 4.3 August Timing Schemes

For the month of August, 86 days were used as the verification set. The starting times were produced as previously mentioned and then compared to the actual starting time. The average mean error and standard deviation can be seen in Table 7.

	NP with NT	LRP with NT	NP with LRTC	LRP with LRTC	Avg Tstorm Time
MEAN	- 47 min	-1 hr 6 min	-54 min	-24 min	-3 min
STD DEV	4 hr 6 min	3 hr 52 min	3 hr 2 min	3 hr 15 min	2 hr 35 min

**Table 7 Mean and Standard Deviation of Error between Actual Starting Time and Forecast Starting Time for August**

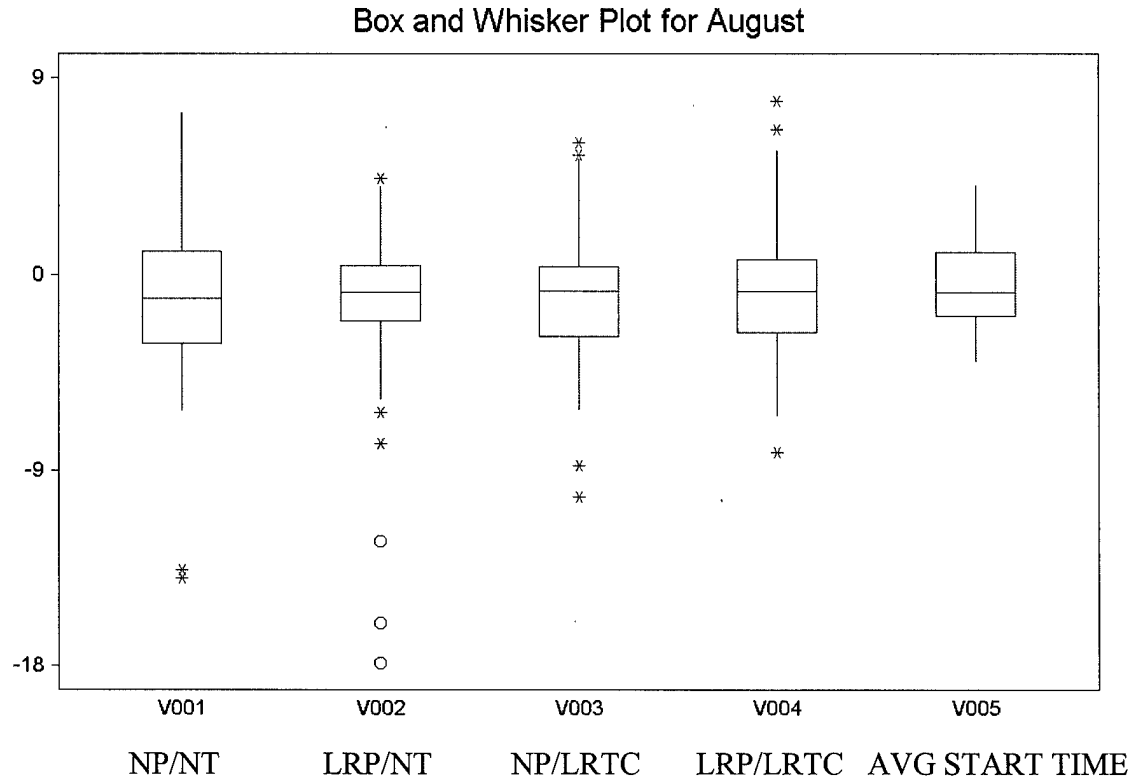
In August, it was found that the average start time performed the best as far as the standard deviation and mean error are concerned. The normal plot for August can be found in Figure 11.



**Figure 11 Graph of Mean and Standard Deviation Error from Actual Starting Time for August**

This plot shows that indeed, using the average thunderstorm start time produces a tighter standard deviation about the mean. Also, the height of the mean is much higher than all other methods examined.

The box and whisker chart for this data set can be seen in Figure 12.



**Figure 12 Box and Whiskers Plot of Error Data for August**

This plots gives more insight into which is the best method to use for August. Of note, is the number of outliers for all methods examined. The large number of outliers for all algorithms shows that these methods have a higher chance of producing an incorrect forecast whereas the average thunderstorm start time produces no outliers. As can be seen above, the average thunderstorm start time produces small whiskers and no outliers indicating a very good chance of being closer to the actual thunderstorm start time than the other methods.

When forecasting thunderstorm starting times in August, the average thunderstorm start time should be used. Using the average start time will guarantee a closer forecast start time to the actual thunderstorm compared to the other methods.

#### 4.4 September Timing Schemes

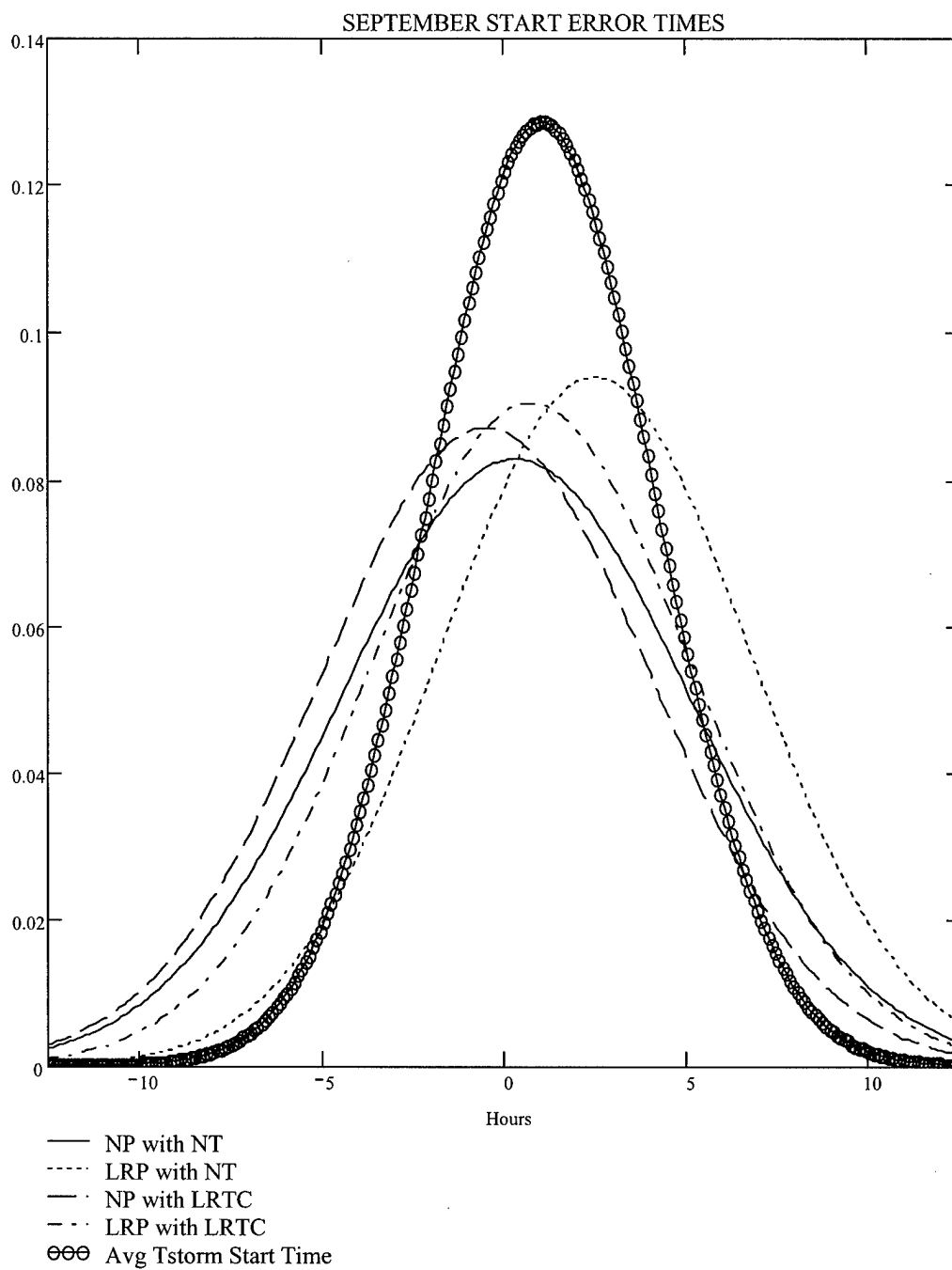
For the month of September, 31 days were used as the verification set. The average mean error and standard deviation can be seen in Table 8.

	NP with NT	LRP with NT	NP with LRTC	LRP with LRTC	Avg Tstorm Time
MEAN	49 min	2 hr 33 min	-37 min	33 min	-3 min
STD DEV	5 hr 23 min	4 hr 10 min	4 hr 30 min	4 hr 23 min	3 hr 6 min

**Table 8 Mean and Standard Deviation of Error between Actual Starting Time and Forecast Starting Time**

In the month of September, there appear to be some larger inconsistencies between methods. For one, the LRP with NT method has a very large mean error when compared to the other methods. Also, the NP with LRTC and LRP with LRTC methods have somewhat similar standard deviations but with much better mean error. However, the average thunderstorm start time performs better than all methods examined both in mean error and standard deviation.

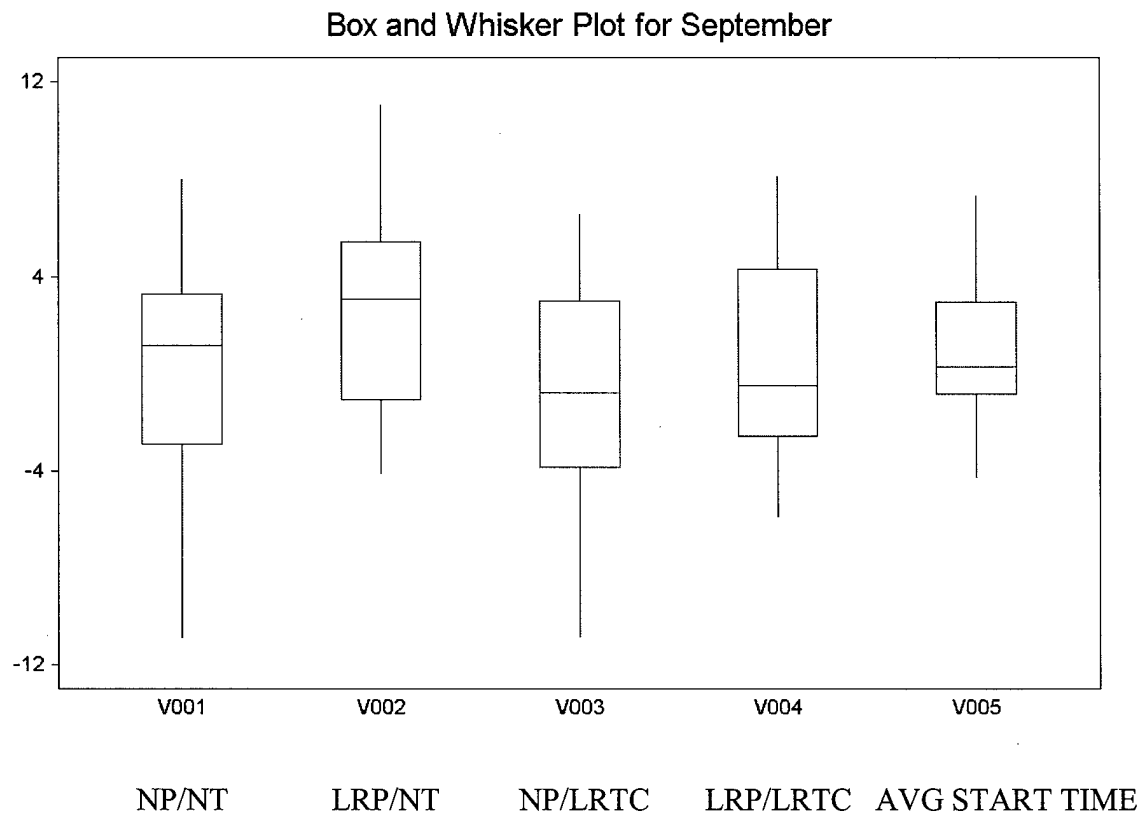
The normalized plot of this data can be found in Figure 13.



**Figure 13 Graph of Mean and Standard Deviation Error from Actual Starting Time for September**

This plot clearly shows how much the LRP with NT's average changes the location of the plot. The large mean error causes the curve to be plotted further towards the positive

x-axis. This plot also shows the similarities in standard deviations between the NP with LRTC and LRP with LRTC. Once again, the height of the plot using average thunderstorm start time is much higher due to the standard deviation. The box and whisker chart for this data set can be seen in Figure 14.



**Figure 14 Box and Whisker Plot for September**

Figure 14 shows some differences between the four methods. First, when NP with NT and the NP with LRTC methods are used a negatively symmetric plot around the median is produced. This indicates that the forecast start times will predict thunderstorm start times many hours earlier than when they actually occur. When using LRP with NT or LRP with LRTC the whiskers are somewhat smaller but the box is so large that no

conclusion can be reached as to which method to use. However, when examining the average thunderstorm start time boxplot, a smaller box is produced with smaller whiskers indicating these times are closer to the actual thunderstorm start time.

The average thunderstorm start time should be used when forecasting thunderstorm start times for September. All tables and figures indicate that using the average start time will produce times that are more accurate.

## **5. Conclusions and Recommendations**

This thesis research was conducted to determine if it was possible to increase the accuracy of forecast thunderstorm start time for Cape Canaveral, Florida. The NPTI algorithm was used as a baseline to compare four methods of forecasting thunderstorm start time. The main concern was to decrease the standard deviation of the forecast start times. Neumann (1971) stated that when using his scheme, a thunderstorm should occur within  $\pm 1\frac{1}{2}$  hours of the forecast start time. The original goal was to decrease this error factor. After using Neumann's index, creating three new ones, and applying the average thunderstorm start time, it was found using the NPTI to forecast thunderstorm start time is highly suspect.

### **5.1 Conclusions**

NPTI was used as a baseline to compare the thunderstorm start times it produced with four methods. Logistic regression was incorporated into the probability forecast of thunderstorms occurring and then applied while using Neumann's prior calculated timing coefficients. Another method incorporated logistic regression in the probability forecast while new timing coefficients were created. A third method took Neumann's method of finding probability and using the newly created timing coefficients to create start times. Finally, the average thunderstorm start time was calculated and used to compare thunderstorm start times. Each month had five different results for starting time. It should be noted that the standard deviations for all methods were quite large and may not be as operationally useful as hoped. The average thunderstorm start time outperformed every method, including the NPTI, for each month. The fact that the

NPTI performs worse than when using the average thunderstorm start time indicates that the NPTI is useless when forecasting thunderstorm start times and should not be used by the 45<sup>th</sup> WS.

## **5.2 Recommendations**

It is recommended that a different method be used to calculate thunderstorm start time. From this thesis, it has been found that by using the average thunderstorm start time to forecast the next thunderstorm occurrence produces the smallest standard deviation of error. If no other method is applied, members of the 45<sup>th</sup> WS can calculate the average start time using their much larger data set and use this as a forecast thunderstorm start time. Above all else, the NPTI should no longer be used to forecast thunderstorm start times.

## **5.3 Suggestions for Further Research**

One way to improve upon the average thunderstorm starting time as a predictor for the next thunderstorm occurrence is to continually update the average. That is, have a spreadsheet with all thunderstorm start times included and update the spreadsheet each day with the thunderstorm start times that occurred that day. Obviously, the thunderstorm start time average will not change much from day to day but it will be current for the next forecast period. Furthermore, these average thunderstorm start times could be split into different times of day. For instance, all thunderstorms that occurred between 0600 EST and 1200 EST could be averaged, 1201 EST and 1800 EST could be

averaged and so on. The forecaster then only needs to decide if a thunderstorm will occur, forecast what time frame and use the appropriate thunderstorm start time average.

Another way to produce timing of thunderstorm occurrence would be to examine persistence. From the literature review, it was found that specific types of days cause different timing and spatial distributions of thunderstorms. Therefore, realizing the synoptic pattern over Florida does not change drastically from day to day and the weather pattern today was very similar to what occurred yesterday, the time of thunderstorm occurrence yesterday can be used to forecast today's thunderstorm start time. This method appears to the author to be the next logical step in producing a method to accurately forecast thunderstorm start times.

# **APPENDIX A** **CONSTANTS FOR NPTI**

MAY	JUNE	JULY	AUGUST	SEPTEMBER
F(X1)	F(X1)	F(X1)	F(X1)	F(X1)
0.1787416000000	0.3326784000000	0.4307867000000	0.3627524000000	0.2816768000000
0.0107402000000	0.0217243800000	0.0436669700000	0.0327221100000	0.0125651800000
0.0136565100000	0.0216295000000	0.0105547500000	0.0108520700000	0.0058043300000
0.0004523660000	0.0003762057000	-0.0000398328100	-0.0000562318700	0.0001096534000
-0.0001802959000	-0.0006835819000	-0.0003116464000	0.0010389140000	-0.0002671096000
0.0003397791000	0.0002579025000	-0.0018889460000	-0.0003726890000	0.0000146929100
-0.0000105183800	0.0000011790030	-0.0000561663000	-0.0000335472600	-0.0000109952000
-0.0000395436500	0.0000014379340	0.0000775770300	-0.0001055251000	0.0000029256110
0.0000337641000	-0.0000337377000	-0.0000541738000	-0.0000067723910	0.0000032287110
0.0000016774350	-0.0000219971000	0.0000351905200	0.0000160676300	-0.0000033257030
F(X2)	F(X2)	F(X2)	F(X2)	F(X2)
0.1206249000000	0.2927881000000	0.4145883000000	0.3932798000000	0.2527479000000
0.0108646000000	0.0263845000000	0.0316634000000	0.0311971900000	0.0108420400000
0.0100196400000	0.0102330700000	-0.0007151263000	0.0025457310000	0.0031367860000
0.0002794513000	0.0003206672000	0.0005390949000	0.0001592548000	0.0001899334000
-0.0001012098000	0.0000705507000	0.0000425100900	0.0000966280900	-0.0002175208000
0.0001964561000	0.0001576005000	-0.0000509110800	0.0000288785200	-0.0000354789100
-0.0000019293880	-0.0000309031700	-0.0000242554500	-0.0000374513500	-0.0000054498940
-0.0000109538900	-0.0000142248900	0.0000158115900	-0.0000171733700	-0.0000044273360
-0.0000065125540	0.0000055886060	-0.0000217213400	-0.0000170416500	0.0000061225120
-0.0000019319070	-0.0000092254160	-0.0000106090400	0.0000040829210	0.0000054122320
F(X3)	F(X3)	F(X3)	F(X3)	F(X3)
0.1037449000000	0.1350110000000	-0.1029031000000	2.5624930000000	0.1736004000000
-0.0119685400000	-0.0199929100000	-0.0029067590000	-0.1702073000000	-0.0191829100000
0.0004832994000	0.0008150658000	0.0004229306000	0.0035513890000	0.0006220711000
-0.0000035704430	-0.0000063425780	-0.0000033083010	-0.0000216134100	-0.0000044144120
F(X4)	F(X4)	F(X4)	F(X4)	F(X4)
0.4273235000000	0.6102192000000	0.6177575000000	0.5271789000000	0.4078606000000
-0.0748021000000	-0.0806676100000	-0.0642101800000	-0.0353019900000	-0.0637667800000
0.0030567000000	0.0024037560000	0.0013104110000	-0.0010948830000	0.0025719610000
F(X5)	F(X5)	F(X5)	F(X5)	F(X5)
-0.5430778000000	-0.1323037000000	0.9355280000000	-0.4163536000000	3.7580340000000
0.0068556030000	0.0010708580000	-0.0037718160000	0.0139472400000	-0.0228789000000
-0.0000105370700	0.0000120896200	0.0000069185940	-0.0000449318900	0.0000359878500
Final Probability Coefficients	Final Probability Coefficients	Final Probability Coefficients	Final Probability Coefficients	Final Probability Coefficients
-0.1589600000000	-0.5556200000000	-0.5553800000000	-0.4623000000000	-0.6183000000000
-0.5503100000000	0.6102500000000	-0.6370500000000	-0.6391600000000	-0.5269300000000
0.3738200000000	0.4851800000000	0.4154200000000	0.4061400000000	0.6065500000000
0.3233200000000	0.3646000000000	0.4982000000000	0.4244200000000	0.5539000000000
0.5656900000000	0.3541600000000	0.4217900000000	0.5676600000000	0.4831500000000
0.0205300000000	0.6391500000000	0.2361400000000	0.0606200000000	1.2949100000000

**APPENDIX B**  
**NEUMANN TIME COEFFICIENTS**  
**( $C_1, C_2, \dots, C_{35}$  in Equation (12))**

To be Used for Each Month
12.7383100000000
-55.2478500000000
16.7874300000000
-3.0446580000000
0.1428297000000
0.4120300000000
-0.0598436800000
-0.0010206330000
-0.0008332961000
0.0000020103010
1.0912470000000
-0.2395890000000
0.8079287000000
-0.0097428530000
-0.0033307730000
0.0000257237800
0.0032750300000
-0.0000011280650
0.0000392877200
-0.0003878205000
0.0385699100000
0.2283849000000
-1.1152510000000
-0.0013180920000
0.0057003430000
-0.0000002110866
0.0052097110000
0.0000525338400
0.0000045646360
-0.0006700146000
0.0126041900000
0.0099107550000
-0.0001220748000
0.0001705095000
-0.0000491474000

## APPENDIX C

### INTERPOLATION PROGRAM USING IDL<sup>®</sup>

Pro interp

; This program will interpolate missing data from upper-air soundings  
; First, the number of lines in the sounding must be calculated

n = 0

s = ' '

close, 5

openr, 5, 'UAJUNE.txt' ; This opens my upperair data for  
; June  
whilenot (eof)) do begin ; If the end of the file has not  
; been reached  
; start reading data lines

readf, 5, s ; Read the file and also number of  
; spaces

if (strlen(s) GT 5) then n = n + 1 ; If the length of spaces exceeds 5  
; then that data line is finished.  
; Read next line

endwhile

; Read in the sounding data

data = fltarr(11,n)  
readf, 5, data  
close, 5

; The next lines give each column of the new array a name

time = data[0,\*]  
day = data[1,\*]  
month = data[2,\*]  
year = data[3,\*]  
pres = data[4,\*]  
hgt = data[5,\*]  
temp = data[6,\*]  
dp = data[7,\*]  
dir = data[8,\*]  
spd = data[9,\*]  
rh = data[10,\*]

; The following line identifies the missing values of temperature

blanks = Where(strpos(temp,'999.0') GE 0, bc) ; This gives row numbers  
; where 999 is reported

nonblanks = Where (strpos(temp,'999.0') LT 0, nbc) ; This gives all row

```

; numbers where temp
; is reported

;The following for loop finds the number given before and after a 999
;is reported

for i = 0L, bc-1 do begin
    before = max(where(nonblank LT blanks(i))) ; This finds the first
                                                ; number given before a
                                                ; 999 is reported

    after = min(where(nonblank GT blanks(i))) ; This finds the first
                                                ; number after a 999
                                                ; is reported

; The following equation calculates the missing value

temp(blanks(i)) = temp(nonblanks(before)) + ((temp(nonblanks(after)) - $
temp(nonblanks(before))) * (float(blanks(i) - nonblanks(before))/float $
(nonblank(after) - nonblanks(before))))

endfor

;The following lines use the same method to find the missing dewpoints

blanks = Where(strpos(dp,'999.0') GE 0, bc)
nonblanks = Where(strpos(dp,'999.0') LT 0, nbc)

for i = 0L, bc-1 do begin
    before = max(where(nonblank LT blanks(i)))
    after = min(where(nonblank GT blanks(i)))

dp(blanks(i)) = dp(nonblanks(before)) + ((dp(nonblanks(after)) - $
dp(nonblanks(before))) * (float(blanks(i) - nonblanks(before))/float $
(nonblank(after) - nonblanks(before))))

endfor

;To find the missing RH values Teton's Formula was used

norh = Where(strpos(rh,'999.0') GT 0 ,bc)

for i = 0L, bc-1 do begin

rh(norh(i)) =
100*(6.112*EXP((17.67*dp(i))/dp(i)+243.5)))/(6.112*EXP((17.67$
*temp(i))/(temp(i)+243.5)))

endfor

; Now a new array is formed with all interpolated values

array = [time,day,month,year,pres,hgt,temp,dp,dir,spd,rh]
;The following lines output the array to a new file

```

```

openu, outfile, "juninterped.txt", /get_lun
form=' (5.0,2x,f5.0,2x,f5.0,2x,f5.0,2x,f5.0,2x,f5.0,2x,f5.0,2x,f
5.0,2x,f5.0,2x,f5.0,2x)'

for i =0,n-1 do begin
printf,outfile,time(i),day(i),month(i),year(i),pres(i),hgt(i),temp(i),d
p(i),$
dir(i),spd(i),rh(i), format = form
endfor

close, outfile

free_lun, outfile

end

```

**APPENDIX D**  
**QBASIC® PROGRAM FINDS NEAREST 50MB INCREMENT**  
**THIS DATA IS THEN USED IN THE PROGRAM GIVEN IN APPENDIX D**

```
'Initial Variables
InputFileName$ = "C:\Thesis\UA\Juntext.txt": 'Read from file
OutputFileName$ = "C:\newjunua.txt": 'Output to file

TotalNumberOfLinesReadSoFar = 0
FileHasBeenExhausted$ = "False"

'Clear The Screen
CLS

'Open Files For Input And Output

OPEN InputFileName$ FOR INPUT AS #1
OPEN OutputFileName$ FOR OUTPUT AS #2

DO UNTIL FileHasBeenExhausted$ = "True"

    'Read Through The Lines Read So Far.
    'Must Close And Open The File, So That
    'Reading Begins At The Beginning.

    CLOSE 1
    OPEN InputFileName$ FOR INPUT AS #1
    FOR ReadThroughTheLines = 1 TO TotalNumberOfLinesReadSoFar

        'These Are Just Dummy Numbers That Are Ignored.
        'This Is Being Done To Get Through The File

        INPUT #1, Hour, Day, Month, Year, Pressure, Height,
        Temperature, DewPoint, WindDirection, WindSpeed,
        RelativeHumidity

    NEXT

    'Find Out How Many Rows There Are Of The Same Time And Date.
    'First, Read In What Is Going To Be Matched Against.
    'Note: The 'F' In Front Of Each Variable Name Abbreviation for
    'First Of It's Kind

    INPUT #1, FHour, FDay, FMonth, FYear, FPressure, FHeight,
    FTemperature, FDewPoint, FWindDirection, FWindSpeed,
    FRelativeHumidity

    'Now Loop Until No Match For Time, Day, Month And Year Is Found,
    'Keeping Count Of The Number Of Matching Rows.

    NumberOfMatchingRows = 0
    DO
        IF EOF(1) THEN FileHasBeenExhausted$ = "True": EXIT DO
        INPUT #1, Hour, Day, Month, Year, Pressure, Height,
```

```

    Temperature, DewPoint, WindDirection, WindSpeed,
    RelativeHumidity
    NumberOfMatchingRows = NumberOfMatchingRows + 1

LOOP WHILE (FHour = Hour AND FDay = Day AND FMonth = Month AND
           FYear = Year)

    'Now Build Arrays That Are Large Enough To Hold
    'The Data For This Matching Set Of Times And Dates
    'These Next 3 Lines Are For The One Instance When The Last Line
    'Of The Input File Has Just Been Hit.

    IF FileHasBeenExhausted$ = "True" THEN
        NumberOfMatchingRows = NumberOfMatchingRows + 1
    END IF

    'Build The Arrays. Note: REDIM Must Be Used When Arrays Are
    'Going To Be Resized In The Middle Of A Program, Otherwise DIM
    'Is Used

REDIM HR(NumberOfMatchingRows)
REDIM DY(NumberOfMatchingRows)
REDIM MO(NumberOfMatchingRows)
REDIM YR(NumberOfMatchingRows)
REDIM PR(NumberOfMatchingRows)
REDIM HT(NumberOfMatchingRows)
REDIM TP(NumberOfMatchingRows)
REDIM DP(NumberOfMatchingRows)
REDIM WD(NumberOfMatchingRows)
REDIM WS(NumberOfMatchingRows)
REDIM RH(NumberOfMatchingRows)

    'Populate The Arrays With The Data
    'But First Read Through The Lines Read So Far.
    'This Must Be Done Because The Program Has Already Read Through The
    'Data Is To Be Placed In The Array, Must Read Again From The
    'Beginning.

CLOSE 1

OPEN InputFileName$ FOR INPUT AS #1
FOR ReadThroughTheLines = 1 TO TotalNumberOfLinesReadSoFar

    'These Are Just Dummy Numbers That Will Be Ignored.
    'This Is Being Done To Get Through The File

INPUT #1, Hour, Day, Month, Year, Pressure, Height, Temperature,
      DewPoint, WindDirection, WindSpeed, RelativeHumidity
NEXT

    'Now Program Knows How Many Rows There Are Of The Same Time And
    'Date. They Are Used To Populate The Arrays With Data

FOR ReadData = 1 TO NumberOfMatchingRows

```

```

    PRINT "Reading New Data"; ReadData
    INPUT #1, HR(ReadData), DY(ReadData), MO(ReadData),
           YR(ReadData), PR(ReadData), HT(ReadData),
           TP(ReadData), DP(ReadData), WD(ReadData),
           WS(ReadData), RH(ReadData)
NEXT
PRINT "Reading Through Line"; ReadThroughTheLines

'Next Line Counts How Many Lines Have Been Read Through

TotalNumberOfLinesReadSoFar = TotalNumberOfLinesReadSoFar +
                               NumberOfMatchingRows

'Now The Array Holds All The Data With Matching Time And Date

'Time To Start Looking For Matches.

FOR PressureToCheck = 1000 TO 500 STEP -50

    'Find The Closest Pressure In The Array That Is Above And Below
    'The Value Of PressureToCheck.
    'And Also Check For A Perfect Match.
    'But Before Looking, Set Up Initial Values Before Each Pass

    MatchFound$ = "False"
    MatchRow = 1
    ClosestValueAbove = 9999
    RowOfClosestValueAbove = 1
    ClosestValueBelow = 0
    RowOfClosestValueBelow = NumberOfMatchingRows

    'Read The Input File And Compare

    FOR CurrentRowInArray = 1 TO NumberOfMatchingRows

        'Look For A Perfect Match.

        IF PR(CurrentRowInArray) = PressureToCheck THEN
            MatchFound$ = "True"
            MatchRow = CurrentRowInArray
        END IF

        'Check If This Row Is The Closest Value Above Whats
        'Being Looked For.

        IF PR(CurrentRowInArray) - PressureToCheck <
ClosestValueAbove - PressureToCheck AND PR(CurrentRowInArray) >
PressureToCheck THEN
            ClosestValueAbove = PR(CurrentRowInArray)
            RowOfClosestValueAbove = CurrentRowInArray
        END IF

        'Check If This Row Is The Closest Value Below Whats
        'Being Looked For.

```

```

        IF PressureToCheck - PR(CurrentRowInArray) <
PressureToCheck - ClosestValueBelow AND PR(CurrentRowInArray) <
PressureToCheck THEN
            ClosestValueBelow = PR(CurrentRowInArray)
            RowOfClosestValueBelow = CurrentRowInArray
        END IF
    NEXT

    'Write The Rows Needed From The Array To The Output File.
    'This Will Be Either 1 Row If There Is A Perfect Match,
    'Or 3 Rows... The Two Values Above And Below And The Value
    'Itself With 999s In Missing Values.

    IF MatchFound$ = "True" THEN
        PRINT #2, LTRIM$(STR$(HR(MatchRow))) + "," +
LTRIM$(STR$(DY(MatchRow))) + "," + LTRIM$(STR$(MO(MatchRow))) + "," +
LTRIM$(STR$(YR(MatchRow))) + "," + LTRIM$(STR$(PR(MatchRow))) + "," +
LTRIM$(STR$(HT(MatchRow))) + "," + LTRIM$(STR$(TP(MatchRow))) + "," +
LTRIM$(STR$(DP(MatchRow))) + "," + LTRIM$(STR$(WD(MatchRow))) + "," +
LTRIM$(STR$(WS(MatchRow))) + "," + LTRIM$(STR$(RH(MatchRow)))
    ELSE

        'No Match Found So The 'Above'
        'Value Must Be Written In Array

        PRINT #2, LTRIM$(STR$(HR(RowOfClosestValueAbove))) + "," +
LTRIM$(STR$(DY(RowOfClosestValueAbove))) + "," +
LTRIM$(STR$(MO(RowOfClosestValueAbove))) + "," +
LTRIM$(STR$(YR(RowOfClosestValueAbove))) + "," +
LTRIM$(STR$(PR(RowOfClosestValueAbove))) + "," +
LTRIM$(STR$(HT(RowOfClosestValueAbove))) + "," +
LTRIM$(STR$(TP(RowOfClosestValueAbove))) + "," +
LTRIM$(STR$(DP(RowOfClosestValueAbove))) + "," +
LTRIM$(STR$(WD(RowOfClosestValueAbove))) + "," +
LTRIM$(STR$(WS(RowOfClosestValueAbove))) + "," +
LTRIM$(STR$(RH(RowOfClosestValueAbove)))

        'Then The Actual Value With The Time, Day, Month, Year And
        '999s

        PRINT #2, LTRIM$(STR$(HR(RowOfClosestValueAbove))) + "," +
LTRIM$(STR$(DY(RowOfClosestValueAbove))) + "," +
LTRIM$(STR$(MO(RowOfClosestValueAbove))) + "," +
LTRIM$(STR$(YR(RowOfClosestValueAbove))) + "," +
LTRIM$(STR$(PressureToCheck)) + ",999,999,999,999,999,999"

        'Then The 'Below' Value

        PRINT #2, LTRIM$(STR$(HR(RowOfClosestValueBelow))) + "," +
LTRIM$(STR$(DY(RowOfClosestValueBelow))) + "," +
LTRIM$(STR$(MO(RowOfClosestValueBelow))) + "," +
LTRIM$(STR$(YR(RowOfClosestValueBelow))) + "," +
LTRIM$(STR$(PR(RowOfClosestValueBelow))) + "," +
LTRIM$(STR$(HT(RowOfClosestValueAbove))) + "," +
LTRIM$(STR$(TP(RowOfClosestValueBelow))) + "," +

```

```

LTRIM$(STR$(DP(RowOfClosestValueBelow))) + "," +
LTRIM$(STR$(WD(RowOfClosestValueBelow))) + "," +
LTRIM$(STR$(WS(RowOfClosestValueBelow))) + "," +
LTRIM$(STR$(RH(RowOfClosestValueBelow)))
    END IF
NEXT
PRINT "Writing To Output File"

'Go Back And Do It All Again For The Next Different Time And Date
LOOP

PRINT "Total Lines Read"; TotalNumberOfLinesReadSoFar

CLOSE 1
CLOSE 2

END

```

## APPENDIX E

### IDL<sup>®</sup> PROGRAM USED TO FIND MEAN RH

```

Pro meanrh

; This program will calculate the mean relative humidity from 800 to
; 600 mb

n = 0

s = ' '

close, 5

openr, 5, 'UAJUNE.txt'           ; This opens my upperair data for
                                ; June
whilenot (eof)) do begin         ; If the end of the file has not
                                ; been reached
                                ; start reading data lines

readf, 5, s                      ; Read the file and also number of
                                ; spaces
if (strlen(s) GT 5) then n = n + 1 ; If the length of spaces exceeds 5
                                ; then that data line is finished.
                                ; Read next line

endwhile

; Read in the sounding data

data = fltarr(11,n)
readf, 5, data
close, 5

; The next lines give each column of the new array a name

time = data[0,*]
day = data[1,*]
month = data[2,*]
year = data[3,*]
pres = data[4,*]
hgt = data[5,*]
temp = data[6,*]
dp = data[7,*]
dir = data[8,*]
spd = data[9,*]
rh = data[10,*]

keep = Where(pres EQ 800 or pres EQ 750 or pres EQ 700 or pres EQ 650
            or pres EQ 600) ;This line will be used so that only those
                        ;pressure levels needed will be used

```

```

length = n_elements(keep); This tells how long the 'new' data set will be

sum = fltarr(1) ; This makes sum a floating array with one column
l=0
test = data(*,keep) ; This makes an array of only rows as defined by
                    ; keep above

; The following nested loop will average the new data array in
; increments of five. This is done so after 5 averages are performed,
; the loop starts over again

final = fltarr(12,n)

for j = 0, length-1, 5 do begin

sum = 0

    for i = 0,3 do begin
        meanrh = 1/(alog(800) -$
alog(600))*((test(10,j+1)+test(10,j+i+1))/2*(alog(test(4,j+i))-$
alog(test(4,j+i+1))))

        sum = sum + meanrh
    endfor

    for k = 0,10 do begin ; This do loop will put the meanrh value
                        ; into every row that corresponds to that ;
                        ; same day and time in the final array

        final(11,l+k) = sum

    endfor

    l = l + 11 ; counter which ensures the next meanrh value goes in the
                ; correct row

endfor

; The next statements make the final array with all values including
; the mean rh value

final(0,*) = data(0,*)
final(1,*) = data(1,*)
final(2,*) = data(2,*)
final(3,*) = data(3,*)
final(4,*) = data(4,*)
final(5,*) = data(5,*)
final(6,*) = data(6,*)
final(7,*) = data(7,*)
final(8,*) = data(8,*)
final(9,*) = data(9,*)
final(10,*) = data(10,*)

; The next lines output the array to a file

```

```

openu, outfile, "junwithmeanrh.txt", /get_lun,/append

form='(5.0,2x,f5.0,2x,f5.0,2x,f5.0,2x,f5.0,2x,f5.0,2x,f5.0,2x,f
5.0,2x,f5.0,2x,f5.0,2x)'

for i = 0, (n-1)/n do begin
printf, outfile, final, format = form
endfor

close, outfile

free_lun, outfile

end

```

## APPENDIX F

### IDL<sup>®</sup> PROGRAM USED TO FIND SSI

Pro SSI

```
; This program will calculate the Showalter Stability Index

n = 0

s = ' '

close, 5

openr, 5, 'UAJUNE.txt'           ; This opens my upperair data for
                                ; June
whilenot (eof)) do begin         ; If the end of the file has not
                                ; been reached
                                ; start reading data lines

readf, 5, s                      ; Read the file and also number of
                                ; spaces
if (strlen(s) GT 5) then n = n + 1 ; If the length of spaces exceeds 5
                                ; then that data line is finished.
                                ; Read next line

endwhile

; Read in the sounding data

data = fltarr(12,n)
readf, 5, data
close, 5

; The next lines give each column of the new array a name

time = data[0,*]
day = data[1,*]
month = data[2,*]
year = data[3,*]
pres = data[4,*]
hgt = data[5,*]
temp = data[6,*]
dp = data[7,*]
dir = data[8,*]
spd = data[9,*]
rh = data[10,*]
mean = data[11,*]

Cp = .24 ; Specific heat of dry air at constant pressure

C = 273.16 ; 0 degrees Celsius in Kelvin

Epsilon = .05 ; Error margin when temperature at 500 mb
```

```

Keep = Where(pres EQ 850 or pres EQ 500); This keeps pressure
                                         ; levels needed for SSI

test = data(*,keep) ; makes an array of values defined by keep6,j)

*****The following do loop calculates SSI*****
*****Many steps involved*****

length = n_elements(keep)

SSI = fltarr(1) ;SSI will be a floating array with 1 column

final = fltarr(13,n) ;The final array will be 13 by n as read above

for j = 0, length-1,2 do begin
    ssi = 0

;The following finds the temp at the LCL reported in Kelvin

    Tlcl = (test(7,j) - ((.212+.001571*test(7,j)-.000436*$
        test(6,j))*(test(6,j)-test(7,j)))+C)

    T850 = test(6,j)+ C ; 850 mb temp converted to Kelvins

    TD850 = test(7,j) + C; 850 mb dewpoint converted to Kelvins

    T500 = test(6,j+1) + C; 500 mb temp converted to Kelvins

    PLCL = 850*((Tlcl/T850)^(1/.2854)); Pressure level at LCL

;The following determines which form of Teton's formula to use and
;also which equation to find the latent heat of water vapor

    if(Tlcl GE C) then begin

        e = 6.11*10^((7.5*(Tlcl-C))/(237.3+(Tlcl - C)))

        L = (597.3 - (.564*(Tlcl-C)))

    endif else begin

        e = 6.11*10^((9.5*(Tlcl-C))/(265.5+(Tlcl - C)))

        L = (597.3 - (.574*(Tlcl-C)))

    endelse

    Rlcl = ((.62197*e)/(Plcl-e)); Mixing ratio at the LCL

    Thetad = (Tlcl*((850.0/(Plcl-e))^(.2854))); Partial Potential
                                         ; Temperature at LCL

```

```

Thetase = thetad*(exp((L*Rlcl)/(CP*Tlcl))); Psuedo-equivalent
                                           ; Potential Temperature
                                           ; at LCL

TP = (C-5.0) ; This is the estimated value of temperature at
           ; 500 mb.

DeltaT = .05 ; Estimated value of change in T

EP = 6.11*10^((9.5*((TP-C)))/(265.5+((TP-C))); Vapor Pres at 500mb

LP = (597.3 - (.574*(TP-C))); Latent heat of water vapor at 500mb

RP = ((.62197*EP)/(500.0-EP)); Mixing Ratio at 500 mb

ThetaP = TP*((850.0/(500.0-EP))^(.2854)); Partial Potential
                                           ; temperature at 500 mb

Thetaep = ThetaP*(exp((LP*RP)/(CP*TP))); Psuedo-equivalent
                                           ; Potential Temperature at
                                           ; 500 mb

; The following if/then statements are used to get the estimated value
; of the 500 mb temperature as close to zero as possible
; This will give the closest approximation to our actual value

err = (thetaep-thetase)

if(abs(err) LT epsilon) then begin ; if the error is < .05 accept
                                   ; as the true value of 500 mb

TP500 = TP
Goto, jump2 ; Actual value found, goto this line

endif else begin

jump1: TP2 = TP + DeltaT ; value is not <.05. Add value of deltaT and
                       ; calculate again

EP = 6.11*10^((9.5*((TP2-C)))/(265.5+((TP2-C)))

LP = (597.3 - (.574*(TP2-C)))

RP = ((.62197*EP)/(500.0-EP))

ThetaP = TP2*((850.0/(500.0-EP))^(.2854))

Thetaep = ThetaP*(exp((LP*RP)/(CP*TP2)))

Errrp = (thetaep - thetase)

endelse

if(abs(errrp) LT epsilon) then begin; if the error is < .05 accept
                                   ; as actual temperature

```

```

    TP500 = TP2
    Goto,jump2

endif else begin

;The following if/then statements compares the signs of the estimated
;temperature. If they differ in sign, divide delta by 2 and
;recalculate

    if((err LT 0 and errp GT 0) or (err GT 0 and errp lt 0)) then $
    begin

        DeltaT = (.5*(DeltaT))

        Goto,jump1

    endif else begin

;If the signs of the estimated temperatures are the same, compare
;the new estimated temperature with the old

        if(abs(errp) lt abs(err)) then begin

            TP = TP2

            err = errp

            Goto,jump1

        endif else begin
;If the above don't work, make the estimated temperature negative and
;try coming from the opposite direction

            DeltaT = (-1.0*(DeltaT))

            Goto,jump1

        endelse

    endelse

endelse

endelse

; The following statements calculates the SSI

jump2: SSI = (T500 - TP500)

; The following statements put the SSI value in the array

for k = 0,10 do begin

    final(12,y+k) = SSI

endfor

```

```

y=y+11

endfor

final(0,*)=data(0,*)
final(1,*) = data(1,*)
final(2,*) = data(2,*)
final(3,*) = data(3,*)
final(4,*) = data(4,*)
final(5,*) = data(5,*)
final(6,*) = data(6,*)
final(7,*) = data(7,*)
final(8,*) = data(8,*)
final(9,*) = data(9,*)
final(10,*) = data(10,*)
final(11,*) = data(11,*)

; Send to newfile

openu, outfile, "FullJun.txt", /get_lun,/append

form='(5.0,2x,f5.0,2x,f5.0,2x,f5.0,2x,f5.0,2x,f5.0,2x,f5.0,2x,f
5.0,2x,f5.0,2x,f5.0,2x)'

for i = 0, (n-1)/n do begin

printf, outfile, final, format = form

endfor

close, outfile

free_lun, outfile

end

```

## APPENDIX G

### MATHCAD® TEMPLATE TO FIND NPTI

jun = C:/JunFinal.xls

This reads in Upper Air Data

C = C:/Thesis/Constants

Constants that are given in Appendix 1

	1	2	3	4	5	6	7	8	9	10
1	3	11	6	1950	850	1577	18	4	360	6
2	3	11	6	1950	500	5910	-7	-15	23	15
3	3	12	6	1950	850	1536	17	13	338	6
4	3	12	6	1950	500	5856	-8	-18	315	23
5	6	18	6	1950	850	1560	17	13	338	4
6	6	18	6	1950	500	5898	-6	-21	203	8
7	4	20	6	1950	850	1565	17	10	135	6
8	4	20	6	1950	500	5892	-7	-13	270	10
9	3	21	6	1950	850	1576	17	16	180	8
10	3	21	6	1950	500	5938	-6	-11	180	6
11	4	22	6	1950	850	1605	18	15	360	4
12	4	22	6	1950	500	5946	-7	-17	113	4
13	3	25	6	1950	850	1587	20	10	45	2

\*\*\*\* i below depicts 850 mb, j below depicts 500mb. Thus row 1 above is the 850 mb values and row 2 above is the 500 mb values for the SAME sounding. Row 3 & 4 are the NEXT 850 & 500mb values for the next sounding. Month and day below create a new matrix with only values of June and the Day from the above chart. Daynum finds the day number (out of 365 days) that the sounding was taken on.

i := 1, 3.. rows(jun)

j := 2, 4.. rows(jun)

month := submatrix(jun, 1, rows(jun), 3, 3)

day := submatrix(jun, 1, rows(jun), 2, 2)

year := submatrix(jun, 1, rows(jun), 4, 4)

$$\text{daynum}(m, d) := \begin{cases} (120 + d) & \text{if } m=5 \\ (151 + d) & \text{if } m=6 \\ (181 + d) & \text{if } m=7 \\ (212 + d) & \text{if } m=8 \\ (243 + d) & \text{if } m=9 \end{cases}$$

$$\text{DAY}_i := \text{daynum}(\text{month}_i, \text{day}_i)$$

$$\text{DAY} =$$

	1
1	162
2	0
3	163
4	0
5	169
6	0
7	171
8	0
9	172
10	0
11	173

Dir and spd below make 2 separate arrays from the values of the June matrix above. s and t are arrays that find the orthogonal components of the 850mb wind while u and v are arrays that find the orthogonal components of the 500mb wind. RH is the mean RH value for each individual sounding and SS is the Showalter Stability Index from each sounding.

$$\text{dir} := \text{submatrix}(\text{jun}, 1, \text{rows}(\text{jun}), 9, 9)$$

$$\text{spd} := \text{submatrix}(\text{jun}, 1, \text{rows}(\text{jun}), 10, 10)$$

$$s_i := \sin[(\text{dir}_i \cdot 0.0174533) + \pi] \cdot \text{spd}_i$$

$$t_i := \cos[(\text{dir}_i \cdot 0.0174533) + \pi] \cdot \text{spd}_i$$

$$u_j := \sin[(\text{dir}_j \cdot 0.0174533) + \pi] \cdot \text{spd}_j$$

$$v_j := \cos[(\text{dir}_j \cdot 0.0174533) + \pi] \cdot \text{spd}_j$$

$$\text{RH} := \text{submatrix}(\text{jun}, 1, \text{rows}(\text{jun}), 12, 12)$$

$$\text{SS} := \text{submatrix}(\text{jun}, 1, \text{rows}(\text{jun}), 13, 13)$$

$$s =$$

	1
1	-0.0000161569
2	0
3	2.2476254955
4	0
5	1.498416997

$$u =$$

	1
1	0
2	-5.8609693028
3	0
4	16.263417647
5	0

$$t =$$

	1
1	-6
2	0
3	-5.56310881
4	0
5	-3.7087392067

$$v =$$

	1
1	0
2	-13.8075717935
3	0
4	-16.2634942875
5	0

The equations below are what Neumann used to find the probability of a thunderstorm. The constants C are in an 5 X 30 matrix. When k = 1, the constants for May are being used, when k = 2 June is being used and so on to k = 5 when September is being used. X1 = 850mb wind probability, X2 = 500mb wind probability, X3 = RH probability, X4 = SSI probability, and X5 = day number probability.

k := 2

$$X1_i := \left[ C_{1,k} + C_{2,k} \cdot s_i + C_{3,k} \cdot t_i + C_{4,k} \cdot s_i \cdot t_i + C_{5,k} \cdot (s_i)^2 + C_{6,k} \cdot (t_i)^2 + C_{7,k} \cdot (s_i)^3 + C_{8,k} \cdot (s_i)^2 \cdot t_i \right] \dots \\ + C_{9,k} \cdot s_i \cdot (t_i)^2 + C_{10,k} \cdot (t_i)^3$$

$$X2_j := \left[ C_{11,k} + C_{12,k} \cdot u_j + C_{13,k} \cdot v_j + C_{14,k} \cdot u_j \cdot v_j + C_{15,k} \cdot (u_j)^2 + C_{16,k} \cdot (v_j)^2 + C_{17,k} \cdot (u_j)^3 + C_{18,k} \cdot (u_j)^2 \cdot v_j \right] \dots \\ + C_{19,k} \cdot u_j \cdot (v_j)^2 + C_{20,k} \cdot (v_j)^3$$

$$X3_i := C_{21,k} + C_{22,k} \cdot RH_i + C_{23,k} \cdot (RH_i)^2 + C_{24,k} \cdot (RH_i)^3$$

$$X4_i := C_{25,k} + C_{26,k} \cdot SS_i + C_{27,k} \cdot (SS_i)^2$$

$$X5_i := C_{28,k} + C_{29,k} \cdot DAY_i + C_{30,k} \cdot (DAY_i)^2$$

X1 =

	1
1	0.2169369687
2	0
3	0.2624170758
4	0
5	0.2853530899
6	0
7	0.3187585187
8	0

X2 =

	1
1	0
2	0.0862846018
3	0
4	0.5229760776
5	0
6	0.462532137
7	0
8	0.5327847573

X3 =

	1
1	0.1481794391
2	0
3	0.504599384
4	0
5	0.5169495856
6	0
7	0.5086586348
8	0

X4 =

	1
1	0.163329974
2	0
3	0.6102192
4	0
5	0.458499004
6	0
7	0.389850174
8	0

X5 =

	1
1	0.3584552833
2	0
3	0.3634552678
4	0
5	0.3939629388
6	0
7	0.4043255964
8	0

The matrix reg below are the coefficients in the final regression used to find the probability of a thunderstorm at Cape Canaveral. Once again, when k =1 is equivalent to May, etc..

$$\text{reg} := \begin{bmatrix} -.15896 & -.55562 & -.55538 & -.46230 & -.61830 \\ -.55031 & .61025 & -.63705 & -.63916 & -.52693 \\ .37382 & .48518 & .41542 & .40614 & .60655 \\ .32332 & .36460 & .49820 & .42442 & .55390 \\ .56569 & .35416 & .42179 & .56766 & .48315 \\ .02053 & .63915 & .23614 & .06062 & 1.29491 \end{bmatrix}$$

k := 2

$$P_i := \text{reg}_{1,k} + \text{reg}_{2,k} \cdot X1_i + \text{reg}_{3,k} \cdot X2_{i+1} + \text{reg}_{4,k} \cdot X3_i + \text{reg}_{5,k} \cdot X4_i + \text{reg}_{6,k} \cdot X5_i$$

P =	3	0.4906521555
	4	0
	5	0.4455913039
	6	0
	7	0.4793498754
	8	0

The final probability of thunderstorm occurrence is given to the left.

l = C:/NeumannTimeConstants.xls

Reads in Neumann Time constants

Now to find the timing of thunderstorm occurrence, Neumann chose 4 variables. V and W equal the 850mb wind component, X = day number, and y = probability of thunderstorm. These variables are then put in an equation which creates a time.

$$v_i := s_i$$

$$w_i := t_i$$

$$x_i := \text{DAY}_i$$

$$y_i := P_i$$

$$S_i := \left[ l_1 + l_2 \cdot y_i + l_3 \cdot (y_i)^2 + l_4 \cdot (y_i)^3 + l_5 \cdot x_i + l_6 \cdot x_i \cdot y_i + l_7 \cdot x_i \cdot (y_i)^2 + l_8 \cdot (x_i)^2 + l_9 \cdot (x_i)^2 \cdot y_i + l_{10} \cdot (x_i)^3 + l_{11} \cdot w_i \right] \dots$$

$$+ \left[ l_{12} \cdot w_i \cdot y_i + l_{13} \cdot w_i \cdot (y_i)^2 + l_{14} \cdot w_i \cdot x_i + l_{15} \cdot w_i \cdot x_i \cdot y_i + l_{16} \cdot w_i \cdot (x_i)^2 + l_{17} \cdot (w_i)^2 + l_{18} \cdot (w_i)^2 \cdot y_i \right] \dots$$

$$+ \left[ l_{19} \cdot (w_i)^2 \cdot x_i + l_{20} \cdot (w_i)^3 + l_{21} \cdot v_i + l_{22} \cdot v_i \cdot y_i + l_{23} \cdot v_i \cdot (y_i)^2 + l_{24} \cdot v_i \cdot x_i + l_{25} \cdot v_i \cdot x_i \cdot y_i + l_{26} \cdot v_i \cdot (x_i)^2 \right] \dots$$

$$+ \left[ l_{27} \cdot v_i \cdot w_i + l_{28} \cdot v_i \cdot w_i \cdot y_i + l_{29} \cdot v_i \cdot w_i \cdot x_i + l_{30} \cdot v_i \cdot (w_i)^2 + l_{31} \cdot (v_i)^2 + l_{32} \cdot (v_i)^2 \cdot y_i + l_{33} \cdot (v_i)^2 \cdot x_i \right] \dots$$

$$+ l_{34} \cdot (v_i)^2 \cdot w_i + l_{35} \cdot (v_i)^3$$

S =

	1
1	17.1747030406
2	0
3	14.4313738901
4	0
5	14.6343081407
6	0
7	13.5514173529
8	0
9	13.7901898413
10	0

The if/then loop below changes the S output into hours and minutes

$$\text{time}_i := \begin{cases} I_i \leftarrow \text{trunc}(S_i) \\ B_i \leftarrow I_i \\ J_i \leftarrow \text{trunc}[\lceil (S_i - B_i) \cdot 60 \rceil + .5] \\ I_i \cdot 100 + J_i \text{ if } (J_i - 60) < 0 \end{cases}$$

time =

	1
1	1710
2	0
3	1426
4	0
5	1438
6	0
7	1333
8	0
9	1347
10	0
11	1442
12	0
13	1544
14	0
15	1444
16	0

# **APPENDIX H** **LOGISTIC PROBABILITY COEFFICIENTS**

MAY	JUNE	JULY	AUGUST	SEPTEMBER
F(X1)	F(X1)	F(X1)	F(X1)	F(X1)
-2.457421520000000	-0.874825903000000	-0.125785528000000	-0.241378075000000	-0.875096225000000
-0.175571837000000	-0.108233194000000	-0.136059651000000	-0.112923323000000	-0.084504822000000
-0.037791484600000	-0.060554339200000	0.005549849820000	0.005858891230000	-0.019770930900000
0.001085144910000	-0.000866053496000	-0.001265417450000	0.004144994200000	-0.000417157588000
-0.004474436480000	-0.002218383040000	-0.002731524130000	-0.001104244540000	-0.001346995950000
0.001678172440000	0.002074498600000	-0.001842891370000	-0.002275158400000	-0.002230010170000
-0.000037526658500	0.000052330503800	0.000090898567800	-0.000123380071000	0.000102828663000
0.000060294121200	-0.000071290665400	-0.000047764424300	-0.000035896639000	-0.000156920183000
0.000199788826000	0.000027408632400	-0.000116388478000	0.000228944231000	0.000021176450200
-0.000092887354900	0.000065395335700	0.000000436216815	-0.000089218129100	-0.000022163237600
F(X2)	F(X2)	F(X2)	F(X2)	F(X2)
-1.128088720000000	-0.936047050000000	-0.691599188000000	-0.459351617000000	-0.741845629000000
-0.122880696000000	-0.162072255000000	-0.188523902000000	-0.134189574000000	-0.073604312000000
-0.036525350700000	-0.098521851700000	-0.102000033000000	-0.056406428900000	-0.077109490000000
0.001532821280000	-0.004770993980000	-0.000915040098000	0.001508761880000	0.003966308240000
-0.008127147900000	-0.002426121820000	0.000385324800000	0.000813519706000	-0.002899762810000
0.000862859923000	0.000151235824000	-0.002692243840000	-0.001946461610000	-0.002548287530000
-0.000137977701000	0.000186775455000	0.000288908716000	0.000233551088000	0.000107212321000
0.000062199563500	-0.000188589110000	0.000211414286000	0.000030393323200	-0.000110120140000
0.000209652551000	0.000028614979800	-0.000034533031400	-0.000012635211800	0.000170345705000
-0.000089094180300	0.000092777553000	0.000114394286000	0.000048540683500	-0.000006620828770
F(X3)	F(X3)	F(X3)	F(X3)	F(X3)
-7.447942270000000	-3.445040770000000	-3.396935340000000	-3.874789910000000	-19.866129600000000
20.750792000000000	-4.129954090000000	2.665734010000000	6.736364990000000	72.572139600000000
-20.279076100000000	28.902574000000000	8.290612650000000	-3.531710580000000	-92.921926100000000
7.079657170000000	-22.665826700000000	-7.036471510000000	-0.226668992000000	40.859320200000000
F(X4)	F(X4)	F(X4)	F(X4)	F(X4)
-0.572143733000000	-0.087985411000000	-0.074042713000000	-0.237429897000000	-0.708225493000000
-0.265854007000000	-0.223271178000000	-0.211274506000000	-0.197372491000000	-0.300976816500000
-0.019557409700000	-0.033558867900000	-0.008983117750000	0.008716634960000	0.016509598900000
F(X5)	F(X5)	F(X5)	F(X5)	F(X5)
-0.374463884000000	-0.205195233000000	-0.345612993000000	-3.274033240000000	5.703364240000000
0.009375911490000	0.023353022300000	0.010333718300000	0.166762482000000	-0.141837861000000
-0.000202255210000	0.000021921729900	-0.000246551690000	-0.002137242150000	0.000724549114000
Final Probability	Final Probability	Final Probability	Final Probability	Final Probability
Coefficients	Coefficients	Coefficients	Coefficients	Coefficients
-16.6123602	-7.02350958	-5.63431384	-5.13263981	-3.50185959
3.80953959	4.61139118	4.24573142	3.83315244	4.60598495
2.84121773	2.20907802	1.8743895	1.84741373	1.53949334
3.86390824	2.47766016	3.14187661	2.50967823	4.85909819
4.13317458	4.03659792	2.74126511	3.0462098	2.32653736
27.4441936	1.69227621	0.195046844	156.779011	-2.82196658

# **APPENDIX I** **MATHCAD® TEMPLATE TO FIND LRP WITH NT**

Jun= C:\JunFinal.xls      C:\Logicconstants      C is the matrix of coefficients  
to be used in the logistical regression

jun =

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	3	11	6	1950	850	1577	18	4	360	6	39	34	7
2	3	11	6	1950	500	5910	-7	-15	23	15	53	34	7
3	3	12	6	1950	850	1536	17	13	338	6	77	80	0
4	3	12	6	1950	500	5856	-8	-18	315	23	44	80	0
5	6	18	6	1950	850	1560	17	13	338	4	77	62	2
6	6	18	6	1950	500	5898	-6	-21	203	8	29	62	2
7	4	20	6	1950	850	1565	17	10	135	6	63	61	3
8	4	20	6	1950	500	5892	-7	-13	270	10	62	61	3

\*\*\*\* i below depicts 850 mb, j below depicts 500mb. Thus row 1 above is the 850 mb values and row 2 above is the 500 mb values for the SAME sounding. Row 3 & 4 are the 850 & 500mb values for the **NEXT** sounding. Month and day below create a new matrix with only values of the month and the day from the above chart. Daynum finds the day number (out of 365 days) that the sounding was taken on.

i := 1, 3.. rows(jun)

month := submatrix(jun, 1, rows(jun), 3, 3)

j := 2, 4.. rows(jun)

day := submatrix(jun, 1, rows(jun), 2, 2)

daynum(m, d) :=  $\begin{cases} (120 + d) & \text{if } m=5 \\ (151 + d) & \text{if } m=6 \\ (181 + d) & \text{if } m=7 \\ (212 + d) & \text{if } m=8 \\ (243 + d) & \text{if } m=9 \end{cases}$

DAY<sub>i</sub> := daynum(month<sub>i</sub>, day<sub>i</sub>)

DAY =

	1
1	162
2	0
3	163
4	0
5	169
6	0
7	171

Dir and spd below make 2 separate arrays from the values of the jun matrix above. s and t create arrays that find the orthogonal components of the 850mb wind while u and v are arrays that find the orthogonal components of the 500mb wind. RH is the mean RH value for each individual sounding and SS is the Showalter Stability Index from each sounding.

dir := submatrix(jun, 1, rows(jun), 9, 9)  
spd := submatrix(jun, 1, rows(jun), 10, 10)

$s_i := \sin[(dir_i \cdot 0.0174533) + \pi] \cdot spd_i$

$u_j := \sin[(dir_j \cdot 0.0174533) + \pi] \cdot spd_j$

$t_i := \cos[(dir_i \cdot 0.0174533) + \pi] \cdot spd_i$

$v_j := \cos[(dir_j \cdot 0.0174533) + \pi] \cdot spd_j$

RH := submatrix(jun, 1, rows(jun), 12, 12)  
SS := submatrix(jun, 1, rows(jun), 13, 13)

s =

1	-0.000016156922486
2	0
3	2.24762549552182
4	0
5	1.49841699701455
6	0
7	-4.24263640286608
8	0
9	0.000010771281657

+

t =

1	-5.99999999997825
2	0
3	-5.56310880999826
4	0
5	-3.70873920666551
6	0
7	4.24264497136817
8	0
9	7.99999999999275

The equations below are the way Everitt used to find the probability of a thunderstorm. First, he regressed each variable separately to find  $X1...X5$ . Then he logistically regressed these variables to find a new probability that is constrained between 0 and 1. The constants  $C$  are in a  $5 \times 30$  matrix. When  $k = 2$ , the constants for June are being used.  $X1$  = 850mb wind probability,  $X2$  = 500mb wind probability,  $X3$  = RH probability,  $X4$  = SSI probability, and  $X5$  = day number probability.

$$k := 2$$

$$X1_i := C_{1,k} + C_{2,k} \cdot s_i + C_{3,k} \cdot t_i + C_{4,k} \cdot s_i \cdot t_i + C_{5,k} \cdot (s_i)^2 + C_{6,k} \cdot (t_i)^2 + C_{7,k} \cdot (s_i)^3 + C_{8,k} \cdot (s_i)^2 \cdot t_i \dots \\ + C_{9,k} \cdot s_i \cdot (t_i)^2 + C_{10,k} \cdot (t_i)^3$$

$$\text{newX1}_i := \frac{e^{X1_i}}{1 + e^{X1_i}} \quad \text{Here is where the Logistic Regression comes in to play}$$

$$X2_j := (C_{11,k} + C_{12,k} \cdot u_j) + C_{13,k} \cdot v_j + C_{14,k} \cdot u_j \cdot v_j + C_{15,k} \cdot (u_j)^2 + C_{16,k} \cdot (v_j)^2 + C_{17,k} \cdot (u_j)^3 \dots \\ + [C_{18,k} \cdot (u_j)^2 \cdot v_j + C_{19,k} \cdot u_j \cdot (v_j)^2 + C_{20,k} \cdot (v_j)^3]$$

$$\text{newX2}_j := \frac{e^{X2_j}}{1 + e^{X2_j}}$$

$$X3_i := C_{21,k} + C_{22,k} \cdot RH_i + C_{23,k} \cdot (RH_i)^2 + C_{24,k} \cdot (RH_i)^3$$

$$\text{newX3}_i := \frac{e^{X3_i}}{1 + e^{X3_i}}$$

$$X4_i := C_{25,k} + C_{26,k} \cdot SS_i + C_{27,k} \cdot (SS_i)^2$$

$$\text{newX4}_i := \frac{e^{X4_i}}{1 + e^{X4_i}}$$

$$X5_i := [C_{28,k} + C_{29,k} \cdot DAY_i + C_{30,k} \cdot (DAY_i)^2]$$

$$\text{newX5}_i := \frac{e^{X5_i}}{1 + e^{X5_i}}$$

The matrix reg below are the coefficients in the final regression used to find the probability of a thunderstorm at Cape Canaveral. As before, the logistically regresseded are linearly regressed to find the final probability. This final probability is then logisitically regressed. Once again, when k =2 is equivalent to June.

$$k := 2$$

$$\text{reg} := \begin{bmatrix} -16.6123602 & -7.02350958 & -5.63431384 & -5.13263981 & -3.50185959 \\ 3.80953959 & 4.61139118 & 4.24573142 & 3.83315244 & 4.60598495 \\ 2.84121773 & 2.20907802 & 1.87438950 & 1.84741373 & 1.53949334 \\ 3.86390824 & 2.47766016 & 3.14187661 & 2.50967823 & 4.85909819 \\ 4.13317458 & 4.03659792 & 2.74126511 & 3.04620980 & 2.32653736 \\ 27.4441936 & 1.69227621 & .195046844 & 156.779011 & -2.82196658 \end{bmatrix}$$

$$P_i := (\text{reg}_{1,k} + \text{reg}_{2,k} \cdot \text{newX1}_i + \text{reg}_{3,k} \cdot \text{newX2}_{i+1} + \text{reg}_{4,k} \cdot \text{newX3}_i + \text{reg}_{5,k} \cdot \text{newX4}_i + \text{reg}_{6,k} \cdot \text{newX5}_i)$$

$$\text{finalP}_i := \frac{e^{P_i}}{1 + e^{P_i}}$$

P =

	1
1	-1.73183479180746
2	0
3	0.526892400730081
4	0
5	-1.27667558343219
6	0
7	-1.57485667580443
8	0
9	-0.633488991211053
10	0
11	0.690815099104199
12	0
13	-0.174291672254547
14	0
15	-1.18191539064362
16	0
17	-1.7442491629805
18	0
19	-2.79673528277125
20	0

finalP =

	1
1	0.150353040192183
2	0
3	0.628758022595704
4	0
5	0.218116644161796
6	0
7	0.171525135995851
8	0
9	0.346719841122936
10	0
11	0.666148225070128
12	0
13	0.456537051230765
14	0
15	0.234707978689377
16	0
17	0.148774015741928
18	0
19	0.057500850237968
20	0

l = C:/NeumannTimeConstants.xls

Reads in Neumann Time constants

To find the timing of thunderstorm occurrence, Neumann chose 4 variables. V and W equal the 850mb wind components, x = day number, and y = probability of thunderstorm. These variables are then put in an equation which finds a number which is equivalent to the time.

$$v_i := s_i$$

$$w_i := t_i$$

$$x_i := \text{DAY}_i$$

$$y_i := \text{finalP}_i$$

$$S_i := \left[ l_1 + l_2 \cdot y_i + l_3 \cdot (y_i)^2 + l_4 \cdot (y_i)^3 + l_5 \cdot x_i + l_6 \cdot x_i \cdot y_i + l_7 \cdot x_i \cdot (y_i)^2 + l_8 \cdot (x_i)^2 + l_9 \cdot (x_i)^2 \cdot y_i + l_{10} \cdot (x_i)^3 + l_{11} \cdot w_i \right] \dots$$

$$+ \left[ l_{12} \cdot w_i \cdot y_i + l_{13} \cdot w_i \cdot (y_i)^2 + l_{14} \cdot w_i \cdot x_i + l_{15} \cdot w_i \cdot x_i \cdot y_i + l_{16} \cdot w_i \cdot (x_i)^2 + l_{17} \cdot (w_i)^2 + l_{18} \cdot (w_i)^2 \cdot y_i \right] \dots$$

$$+ \left[ l_{19} \cdot (w_i)^2 \cdot x_i + l_{20} \cdot (w_i)^3 + l_{21} \cdot v_i + l_{22} \cdot v_i \cdot y_i + l_{23} \cdot v_i \cdot (y_i)^2 + l_{24} \cdot v_i \cdot x_i + l_{25} \cdot v_i \cdot x_i \cdot y_i + l_{26} \cdot v_i \cdot (x_i)^2 \right] \dots$$

$$+ l_{27} \cdot v_i \cdot w_i + l_{28} \cdot v_i \cdot w_i \cdot y_i + l_{29} \cdot v_i \cdot w_i \cdot x_i + l_{30} \cdot v_i \cdot (w_i)^2 + l_{31} \cdot (v_i)^2 + l_{32} \cdot (v_i)^2 \cdot y_i + l_{33} \cdot \left[ (v_i)^2 \cdot x_i \right] \dots$$

$$+ l_{34} \cdot (v_i)^2 \cdot w_i + l_{35} \cdot (v_i)^3$$

S =	1	16.1253246797798
	2	0
	3	13.5940327235802
	4	0
	5	15.6225291917357
	6	0
	7	16.3156784632206
	8	0
	9	15.2413040744336
	10	0
	11	13.5556249682673

The if/then loop below changes the S output into hours and minutes

$$\text{time}_i := \begin{cases} I_i \leftarrow \text{trunc}(S_i) \\ B_i \leftarrow I_i \\ J_i \leftarrow \text{trunc} \left[ \left[ (S_i - B_i) \cdot 60 \right] + .5 \right] \\ I_i \cdot 100 + J_i \quad \text{if } (J_i - 60) < 0 \end{cases}$$

	1
1	1608
2	0
3	1336
4	0
5	1537
time = 6	0
7	1619
8	0
9	1514
10	0
11	1333
12	0

**APPENDIX J**  
**LINEARLY REGRESSED TIME COEFFICIENTS**

JUNE	JULY	AUGUST	SEPTEMBER
-7.09E-03	-1.36E+03	5.77E+03	-1.03E+04
1.60E+03	-1.47E+03	1.03E+03	-1.51E+03
98.216	-113.913	75.703	71.792
-16.902	-1.395	19.952	-28.209
-4.68	23.207	-77.515	120.826
-19.461	14.498	-9.681	11.342
-0.426	0.596	-0.217	-0.323
0.058	-0.129	0.348	-0.472
0.059	-0.036	0.023	-0.021
-1.74E-04	2.38E-04	-5.19E-04	6.14E-04
-17.573	-48.224	77.802	40.377
-3.148	-1.466	15.849	14.122
-0.549	-1.557	-1.126	-1.131
0.21	0.489	-0.688	-0.308
0.024	5.30E-03	-0.068	-0.055
-6.39E-04	-1.24E-03	1.52E-03	5.82E-04
0.082	-0.081	-0.089	-0.367
4.07E-03	0.029	-0.032	-0.033
-5.54E-04	4.29E-04	4.00E-04	1.45E-03
5.11E-04	-4.28E-04	-6.77E-04	1.25E-03
1.053	-47.129	1.109	-111.619
-10.301	11.743	-6.639	76.626
-0.436	0.941	7.49E-03	-6.41
0.012	0.473	-0.018	0.857
0.063	-0.059	0.032	-0.296
-1.06E-04	-1.18E-03	5.84E-05	-1.64E-03
0.124	-1.58E-03	0.405	0.118
0.016	4.71E-03	-0.055	-0.033
-7.76E-04	5.62E-05	-1.71E-03	-4.63E-04
-3.94E-05	2.54E-04	5.17E-05	-9.93E-04
0.148	0.18	-0.16	-0.314
0.027	2.10E-03	1.68E-03	-0.032
-9.09E-04	-9.33E-04	6.87E-04	1.11E-03
-4.86E-04	-1.41E-04	-8.65E-04	8.79E-04
-4.21E-04	-3.49E-04	2.64E-04	-6.39E-04

## APPENDIX K

### MATHCAD® TEMPLATE TO FIND NP WITH LRTC

Pr :=

  
C:\June\Pr.xls

This file is simply Neumann's probability found by using his method described in Appendix J. It was cut and pasted to an Excel Spreadsheet so could be easily read by MathCad.

data :=

  
C:\June\l(std).xls

This file was taken from an Excel spreadsheet. The spreadsheet uses Equation 12 and performs the appropriate calculations to the variables. These are then used to produce the new timing coefficients.

$y := \text{submatrix}(\text{data}, 1, \text{rows}(\text{data}), 36, 36)$

$nx := \text{submatrix}(\text{data}, 1, \text{rows}(\text{data}), 1, 35)$

$l := (nx^T \cdot nx)^{-1} \cdot nx^T \cdot y$

Matrix algebra that performs linear regression

l =

	1
1	-7.093·10 <sup>-3</sup>
2	1.602·10 <sup>3</sup>
3	98.216
4	-16.902
5	-4.68
6	-19.461
7	-0.426
8	0.058
9	0.059
10	-1.739·10 <sup>-4</sup>
11	-17.573
12	-3.148
13	-0.549
14	0.21
15	0.024
16	-6.386·10 <sup>-4</sup>

NEW TIME COEFFICENTS

$z := 1..(\text{datatest})$

$\text{nDIR} := \text{submatrix}(\text{randata}, 1, \text{rows}(\text{randata}), 9, 9)$

$\text{nSPD} := \text{submatrix}(\text{randata}, 1, \text{rows}(\text{randata}), 10, 10)$

$S_z := \sin\left[\left(\text{nDIR}_z \cdot 0.0174533\right) + \pi\right] \cdot \text{nSPD}_z$

$T_z := \cos\left[\left(\text{nDIR}_z \cdot 0.0174533\right) + \pi\right] \cdot \text{nSPD}_z$

$\text{mth} := \text{submatrix}(\text{randata}, 1, \text{rows}(\text{randata}), 3, 3)$

$\text{dy} := \text{submatrix}(\text{randata}, 1, \text{rows}(\text{randata}), 2, 2)$

Same method as described earlier to find variables needed for time

$\text{Daynum}(m, d) := \begin{cases} (120 + d) & \text{if } m=5 \\ (151 + d) & \text{if } m=6 \\ (181 + d) & \text{if } m=7 \\ (212 + d) & \text{if } m=8 \\ (243 + d) & \text{if } m=9 \end{cases}$

$D_z := \text{Daynum}(\text{mth}_z, \text{dy}_z)$

Now to find the timing of thunderstorm occurrence, Neumann chose 4 variables. V and W equal the 850mb wind component, X = day number, and y = probability of thunderstorm. These variables are then put in an equation which creates a time.

$v_z := S_z$

$w_z := T_z$

$x_z := D_z$

$y_z := \text{Pr}_z$

NOTE: L in testtime below are the NEW time coefficients found from above.

$$\begin{aligned} \text{testtime}_z := & \left[ 1 + l_2 \cdot y_z + l_3 \cdot (y_z)^2 + l_4 \cdot (y_z)^3 + l_5 \cdot x_z + l_6 \cdot x_z \cdot y_z + l_7 \cdot x_z \cdot (y_z)^2 + l_8 \cdot (x_z)^2 + l_9 \cdot (x_z)^2 \cdot y_z + l_{10} \cdot (x_z)^3 \right] \dots \\ & + \left[ l_{11} \cdot w_z + l_{12} \cdot w_z \cdot y_z + l_{13} \cdot w_z \cdot (y_z)^2 + l_{14} \cdot w_z \cdot x_z + l_{15} \cdot w_z \cdot x_z \cdot y_z + l_{16} \cdot w_z \cdot (x_z)^2 + l_{17} \cdot (w_z)^2 \right] \dots \\ & + \left[ l_{18} \cdot (w_z)^2 \cdot y_z + l_{19} \cdot (w_z)^2 \cdot x_z + l_{20} \cdot (w_z)^3 + l_{21} \cdot v_z + l_{22} \cdot v_z \cdot y_z + l_{23} \cdot v_z \cdot (y_z)^2 + l_{24} \cdot v_z \cdot x_z \right] \dots \\ & + \left[ l_{25} \cdot v_z \cdot x_z \cdot y_z + l_{26} \cdot v_z \cdot (x_z)^2 + l_{27} \cdot v_z \cdot w_z + l_{28} \cdot v_z \cdot w_z \cdot y_z + l_{29} \cdot v_z \cdot w_z \cdot x_z + l_{30} \cdot v_z \cdot (w_z)^2 \right] \dots \\ & + l_{31} \cdot (v_z)^2 + l_{32} \cdot (v_z)^2 \cdot y_z + l_{33} \cdot (v_z)^2 \cdot x_z + l_{34} \cdot (v_z)^2 \cdot w_z + l_{35} \cdot (v_z)^3 \end{aligned}$$

The if/then loop below changes the S output into hours and minutes

$$\begin{aligned} \text{time}_z := & \begin{cases} I_z \leftarrow \text{trunc}(\text{testtime}_z) \\ B_z \leftarrow I_z \\ J_z \leftarrow \text{trunc} \left[ \left[ (\text{testtime}_z - B_z) \cdot 60 \right] + .5 \right] \\ I_z \cdot 100 + J_z \quad \text{if } (J_z - 60) < 0 \end{cases} \end{aligned}$$

	1
1	1944
2	1409
3	1429
4	1512
5	1344
6	1409
7	1437
time = 8	1404
9	1356
10	1506
11	1519
12	1328
13	1313
14	1432
15	1449
16	1348

**APPENDIX L**  
**LINEARLY REGRESSED TIME COEFFICIENTS USING LOGISTICALLY**  
**REGRESSED PROBABILITIES**

JUNE	JULY	AUGUST	SEPTEMBER
-7.11E-03	-2.11E+03	7.93E+03	-2.10E+04
-49.093	1.65E+03	-392.542	3.14E+03
74.717	-426.481	-458.287	3.30E+03
11.935	18.545	-14.878	-325.863
0.192	31.152	-105.224	243.195
0.525	-15.264	4.886	-28.245
-0.549	1.961	2.177	-11.838
-9.19E-04	-0.153	0.465	-0.939
-1.11E-03	0.035	-0.014	0.062
1.68E-06	2.50E-04	-6.81E-04	1.21E-03
6.126	-50.857	59.032	22.011
6.652	2.971	1.042	1.37
-0.987	1.489	5.12	-16.237
-0.093	0.5	-0.52	-0.164
-0.036	-0.019	-0.016	2.03E-03
3.26E-04	-1.22E-03	1.15E-03	2.98E-04
0.053	0.011	7.27E-04	-0.329
-0.017	0.011	-0.02	-0.177
-3.38E-04	-8.07E-05	-3.59E-06	1.32E-03
8.63E-04	-3.41E-04	-5.50E-04	1.03E-03
23.743	18.959	-18.055	-99.362
-4.899	-13.965	-29.393	28.168
1.903	3.641	-0.293	-2.431
-0.264	-0.169	0.206	0.781
0.021	0.055	0.134	-0.094
7.36E-04	3.84E-04	-5.61E-04	-1.53E-03
0.033	0.141	-0.085	0.15
-0.044	-0.012	0.077	-0.283
-3.97E-05	-5.83E-04	3.94E-04	-5.28E-04
-6.17E-04	6.07E-04	8.89E-04	-8.30E-04
-0.076	-0.082	-0.361	-0.286
0.013	0.104	0.05	0.197
4.63E-04	1.97E-04	1.54E-03	9.40E-04
-8.75E-04	-8.90E-04	-8.55E-04	1.35E-03
-2.14E-04	4.78E-04	8.24E-04	-2.04E-04

## APPENDIX M

### MATHCAD® TEMPLATE TO FIND LRP WITH LRTC

Pr :=

  
C:\June\LogPr.xls

This file is the probabilities found when using logistical regression. It was cut and pasted to an Excel Spreadsheet so could be easily read by MathCad

data :=

  
C:\June\log(std).xls

This file was taken from an Excel spreadsheet. The spreadsheet uses Equation 12 and performs the appropriate calculations to the variables. These are then used to produce the new timing coefficients.

y := submatrix(data, 1, rows(data), 36, 36)

nx := submatrix(data, 1, rows(data), 1, 35)

$$l := \left( \frac{1}{nx \cdot nx} \right)^{-1} \cdot nx^T \cdot y$$

l =

1	-7.114·10 <sup>-3</sup>
2	-49.093
3	74.717
4	11.935
5	0.192
6	0.525
7	-0.549
8	-9.185·10 <sup>-4</sup>
9	-1.109·10 <sup>-3</sup>
10	1.682·10 <sup>-6</sup>
11	6.126
12	6.652
13	-0.987
14	-0.093
15	-0.036
16	3.264·10 <sup>-4</sup>

NEW TIME COEFFICENTS

```

z := 1.. (datatest )
nDIR := submatrix(randata , 1, rows( randata ), 9, 9)
nSPD := submatrix(randata , 1, rows( randata ), 10, 10)
Sz := sin[ (nDIRz · 0.0174533) + π ] · nSPDz
Tz := cos[ (nDIRz · 0.0174533) + π ] · nSPDz
mth := submatrix(randata , 1, rows( randata ), 3, 3)
dy := submatrix(randata , 1, rows( randata ), 2, 2)

```

```

Daynum(m, d) := ⎧ (120+ d) if m=5
                  (151+ d) if m=6
                  (181+ d) if m=7
                  (212+ d) if m=8
                  (243+ d) if m=9

```

```

Dz := Daynum(mthz, dyz)

```

Now to find the timing of thunderstorm occurrence, Neumann chose 4 variables. V and W equal the 850mb wind component, X = day number, and y = probability of thunderstorm. These variables are then put in an equation which creates a time.

```

vz := Sz
wz := Tz
xz := Dz
yz := Prz

```

NOTE: L in testtime below are the NEW time coefficients found from above.

$$\begin{aligned}
\text{testtime}_z := & \left[ l_1 + l_2 \cdot y_z + l_3 \cdot (y_z)^2 + l_4 \cdot (y_z)^3 + l_5 \cdot x_z + l_6 \cdot x_z \cdot y_z + l_7 \cdot x_z \cdot (y_z)^2 + l_8 \cdot (x_z)^2 + l_9 \cdot (x_z)^2 \cdot y_z + l_{10} \cdot (x_z)^3 \right] \dots \\
& + \left[ l_{11} \cdot w_z + l_{12} \cdot w_z \cdot y_z + l_{13} \cdot w_z \cdot (y_z)^2 + l_{14} \cdot w_z \cdot x_z + l_{15} \cdot w_z \cdot x_z \cdot y_z + l_{16} \cdot w_z \cdot (x_z)^2 + l_{17} \cdot (w_z)^2 \right] \dots \\
& + \left[ l_{18} \cdot (w_z)^2 \cdot y_z + l_{19} \cdot (w_z)^2 \cdot x_z + l_{20} \cdot (w_z)^3 + l_{21} \cdot v_z + l_{22} \cdot v_z \cdot y_z + l_{23} \cdot v_z \cdot (y_z)^2 + l_{24} \cdot v_z \cdot x_z \right] \dots \\
& + \left[ l_{25} \cdot v_z \cdot x_z \cdot y_z + l_{26} \cdot v_z \cdot (x_z)^2 + l_{27} \cdot v_z \cdot w_z + l_{28} \cdot v_z \cdot w_z \cdot y_z + l_{29} \cdot v_z \cdot w_z \cdot x_z + l_{30} \cdot v_z \cdot (w_z)^2 \right] \dots \\
& + l_{31} \cdot (v_z)^2 + l_{32} \cdot (v_z)^2 \cdot y_z + l_{33} \cdot (v_z)^2 \cdot x_z + l_{34} \cdot (v_z)^2 \cdot w_z + l_{35} \cdot (v_z)^3
\end{aligned}$$

The if/then loop below changes the S output into hours and minutes

$$\text{time}_z := \begin{cases} I_z \leftarrow \text{trunc}(\text{testtime}_z) \\ B_z \leftarrow I_z \\ J_z \leftarrow \text{trunc}[\lceil (\text{testtime}_z - B_z) \cdot 60 \rceil + .5] \\ I_z \cdot 100 + J_z \text{ if } (J_z - 60) < 0 \end{cases}$$

	1
1	1551
2	1354
3	1120
4	1509
5	1403
6	1628
7	1453
time = 8	1358
9	1518
10	1519
11	1452
12	1504
13	1453
14	1418
15	1517
16	1529

## APPENDIX N

### MATHCAD® TEMPLATE FOR RANDOM VERIFICATION SET

$jun = C:\JunFinal.xls$

$data_{z,13} := \left( \text{submatrix}(jun, \text{newtest}_z, \text{newtest}_z + 1, 13) \right)$  This line pulls out ALL the values associated with the random row number created above

$randata := \begin{cases} \text{new}_{1,13} \leftarrow \text{stack}(data_{1,13}, data_{2,13}) \\ \text{for } i \in 2.. \text{datatest} - 1 \\ \text{new}_{i,13} \leftarrow \text{stack}(\text{new}_{i-1,13}, data_{i+1,13}) \\ \text{new}_{\text{datatest}-1,13} \end{cases}$  This for loop stacks all of the 850 data together. That is ALL variables in row 1 will be stacked above ALL variables in row 143 and so on.

$z := 1.. \text{datatest}$

$test_z := \text{round}(\text{rnd}(\text{rows}(jun)), 0)$  This line lets MathCad randomly pick 74 lines from the jun matrix originally given. It also rounds the number so it is a whole number.

$newtest_z := \begin{cases} (test_z) + 1 & \text{if } \text{mod}\left(\frac{test_z}{2}, 1\right) = 0 \\ test_z & \text{otherwise} \end{cases}$  This is if/then statement ensures that an odd number is being pulled out. To find the time of T-storm occurrence only 850mb u and v wind components, Probability and Day Number are needed. Therefore, an odd number will guarantee a row with 850mb values.

$newtest =$

1	1
2	143
3	435
4	261
5	611

So row 1 is the first row to be extracted, then row 143 from the jun matrix above will be tested.

Once these values were found, they were manually extracted from the upper air data. These rows were then saved to another file. After all regressions and coefficients were found, the verification data set was placed in prior appendices and a forecast starting time was produced.

## BIBLIOGRAPHY

- Blanchard, David O., and Raul E. Lopez, 1985: Spatial patterns of convection in South Florida. *Monthly Weather Review*, **113**, 1282-1299.
- Burpee, R.W., and L.N. Lahiff, 1984: Area-average rainfall variations on sea-breeze days in south Florida. *Monthly Weather Review*, **112**, 510-519.
- Byers, Horace R., and Harriet R. Rodebush, 1948: Causes of thunderstorms of the Florida peninsula. *Journal of Meteorology*, **5**, 275-280.
- Devore, Jay L., 1995: Probability and Statistics for Engineering and the Sciences, 4<sup>th</sup> ed., Pacific Grove: Duxbury Press, 743 pp.
- Duffield, George F., and G.D. Nastrom, 1983: AWS/TR-83/001, *Equations and Algorithms for Meteorological Applications in Air Weather Service*. Air Weather Service, 58 pp. Available from Defense Technical Information Agency.
- Estoque, M.A., 1962: The sea breeze as a function of the prevailing synoptic situation. *Journal of the Atmospheric Sciences*, **19**, 244-250.
- Everitt, James A., 1999: An Improved Thunderstorm Forecast Index for Cape Canaveral, Florida. Unpublished report. The Air Force Institute of Technology, 98 pp.
- Falls, Lee W., William O. Williford, and Michael C. Carter, 1971: Probability distributions for thunderstorm activity at Cape Kennedy, Florida. *Journal of Applied Meteorology*, **10**, 97-104.
- Frank, N.L., P.L. Moore and G.E. Fisher, 1967: Summer shower distribution over the Florida peninsula as deduced from digitized radar data. *Journal of Applied Meteorology*, **6**, 309-316.
- Gentry, Robert C., and Paul L. Moore, 1954: Relation of local and general wind interactions near the sea coast to time and location of air-mass showers. *Journal of Meteorology*, **11**, 507-511.
- Howell, Cindy L., 1998: Nowcasting thunderstorms at Cape Canaveral, Florida, using an improved Neumann-Pfeffer Thunderstorm Index. Unpublished report. The Air Force Institute of Technology, 93 pp.
- Kachigan, Sam K., 1991: Multivariate Statistical Analysis: A Conceptual Introduction. New York: Radius Press, 303pp.

Neter, John., William Wasserman, and Michael H. Kutner, 1983: Applied Linear Regression Models. Illinois: Richard D. Irwin, Inc., 547 pp.

Neumann, Charles J., 1968: Frequency and duration of thunderstorms at Cape Kennedy, Part I. Weather Bureau Technical Memorandum SOS-2.

—, 1970: Frequency and duration of thunderstorms at Cape Kennedy, Part II. Weather Bureau Technical Memorandum SOS-6.

—, 1971: Thunderstorm forecasting at Cape Kennedy, Florida, utilizing multiple regression techniques. NOAA Technical Memorandum NWS SOS-8.

Nicholls, Melville E., Roger A. Pielke, and William R. Cotton, 1991: A two-dimensional numerical investigation of the interaction between sea breezes and deep convection over the Florida peninsula. *Monthly Weather Review*, **119**, 298-323.

Pielke, R.A., and W.R. Cotton, 1977: A mesoscale analysis over south Florida for a high rainfall event. *Monthly Weather Review*, **105**, 343-362.

Reed, Jack W., 1979: Cape Canaveral sea breezes. *Journal of Applied Meteorology*, **18**, 231-235.

Roeder, William P., Chief of Operations Support Flight and Science and Technical Training Officer for 45<sup>th</sup> Weather Squadron, Patrick Air Force Base, Fl. Briefing, October, 1999. Personal Interview.

Wilks, D.S., 1995: Statistical Methods in the Atmospheric Sciences. San Diego: Academic Press, 467pp.

## VITA

Thomas G. Renwick was born in January 1972 in Bedford, England. He graduated from The Pennsylvania State University in 1993 with a Bachelor of Science degree in Meteorology. After becoming an American citizen in 1995, he applied and was accepted to Officer Training School. Upon graduation and commissioning, he was assigned to the 10<sup>th</sup> Mountain Division at Fort Drum, New York where he was a Staff Weather Officer. His second assignment was to Osan AB in South Korea. He was the assistant flight chief for the DOW, 607<sup>th</sup> Combat Operations Squadron. In August, 1998 he was assigned to the Air Force Institute to Technology. His next assignment will be as a Weather Officer at the Air Force Combat Climatology Center.

Permanent Address: 5N924 Kettlehook Court  
St. Charles, IL 60175

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE March 2000		3. REPORT TYPE AND DATES COVERED Master's Thesis
4. TITLE AND SUBTITLE Timing of Thunderstorm Occurrence for Cape Canaveral, Florida			5. FUNDING NUMBERS	
6. AUTHOR(S) Thomas G. Renwick, Captain, USAF				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 P Street, Building 640 WPAFB, OH 45433-7765			8. PERFORMING ORGANIZATION REPORT NUMBER  AFIT/GM/ENP/00M-10	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) 45th Weather Squadron Attn: Mr. William P. Roeder 1201 Edward H. White II St., MS 7302 Patrick AFB, FL 32925-3238 DSN: 467-8410			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES Advisor: Lt Col Cecilia A. Miner, ENP, DSN: 785-3636, ext. 4645				
12a. DISTRIBUTION AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) This research is concerned with improving an existing algorithm to accurately forecast thunderstorm starting time for Cape Canaveral, Florida. This was accomplished by investigating different linear regression techniques than those used in the existing algorithm. The result is three new thunderstorm start time algorithms. The forecast start times of these new algorithms were then compared to actual thunderstorm start times to determine which method produced the most accurate results. The average thunderstorm starting time was also calculated from the data. This time was also compared to actual thundestorm starting time. Upon examination of the various start times produced, it was found that all algorithms, including the original algorithm, performed worse than using the average thunderstorm start time.				
14. SUBJECT TERMS Thunderstorm, Timing of thunderstorm, Florida, Sea Breeze, Thunderstorm occurrence, Forecasting, Meteorology			15. NUMBER OF PAGES 99	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL	